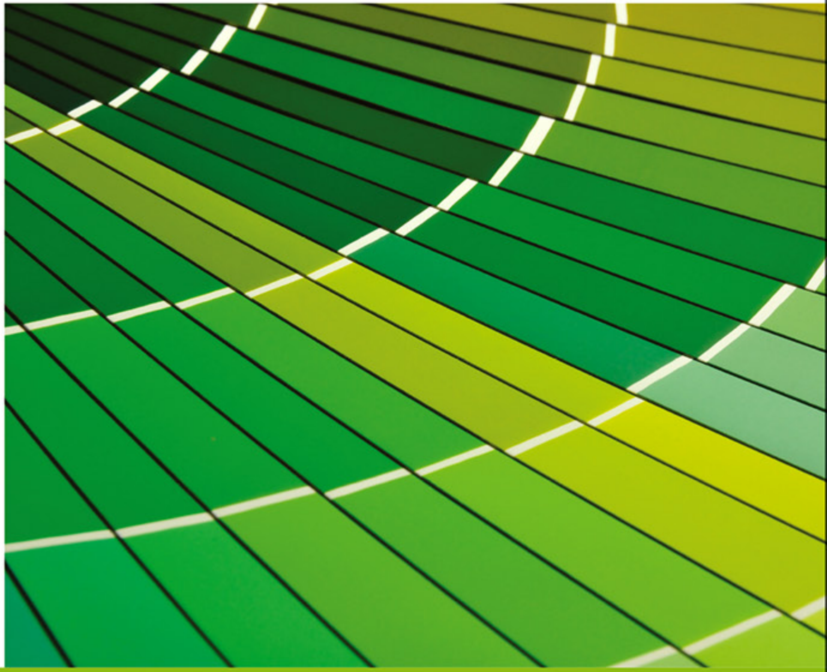
Data-driven
design, Hum
machine int
Design thin
Engineering
Agile, Stor
Personas, P



Usability and User Experience Design

The Comprehensive Guide to Data-Driven UX Design

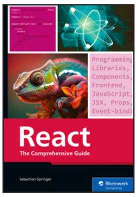
Benjamin Franz
Michaela Kauer-Franz

Rheinwerk
Computing

Rheinwerk Computing

The Rheinwerk Computing series offers new and established professionals comprehensive guidance to enrich their skillsets and enhance their career prospects. Our publications are written by the leading experts in their fields. Each book is detailed and hands-on to help readers develop essential, practical skills that they can apply to their daily work.

Explore more of the Rheinwerk Computing library!


React
The Comprehensive Guide

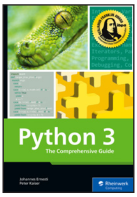
Sebastian Springer
React: The Comprehensive Guide
2024, 676 pages, paperback and e-book
www.rheinwerk-computing.com/5705

Spring Boot 3 and
Spring Framework 6

Christian Ullenboom
Spring Boot 3 and Spring Framework 6
2024, 934 pages, paperback and e-book
www.rheinwerk-computing.com/5764

JS
JavaScript
The Comprehensive Guide

Philip Ackermann
JavaScript: The Comprehensive Guide
2022, 1292 pages, paperback and e-book
www.rheinwerk-computing.com/5554


Python 3
The Comprehensive Guide

Johannes Ernesti, Peter Kaiser
Python 3: The Comprehensive Guide
2022, 1036 pages, paperback and e-book
www.rheinwerk-computing.com/5566

Git
Project Management for Developers and DevOps Teams

Bernd Öggl, Michael Kofler
Git: Project Management for Developers and DevOps Teams
2023, 407 pages, paperback and e-book
www.rheinwerk-computing.com/5555

www.rheinwerk-computing.com

Benjamin Franz, Michaela Kauer-Franz

Usability and User Experience Design

The Comprehensive Guide to Data-Driven UX Design

Rheinwerk
Computing

Imprint

This e-book is a publication many contributed to, specifically:

Editor Megan Fuerst
Acquisitions Editor Hareem Shafi
German Edition Editor Almut Poll
Translation Dr. Benjamin Franz, Fiona Lynch
Copyeditor Yvette Chin
Cover Design Graham Geary
Photo Credit iStockphoto: 91509507/© LarryHerfindal; Shutterstock: 1062413906/ © elenabsl, 394532773/© i-ro
Layout Design Vera Brauner
Production E-Book Graham Geary
Typesetting E-Book SatzPro, Germany

We hope that you liked this e-book. Please share your feedback with us and read the Service Pages to find out how to contact us.

Library of Congress Cataloging-in-Publication Control Number: 2023050539

ISBN 978-1-4932-2496-8 (print)
ISBN 978-1-4932-2497-5 (e-book)
ISBN 978-1-4932-2498-2 (print and e-book)

1st edition 2024
1st German edition published 2022 by Rheinwerk Verlag, Bonn, Germany

Contents

Foreword 21

1 Introduction 23

1.1 Who Is This Book Written For? 23

1.2 Why Should You Read This Book? 24

1.3 How Should You Read This Book? 27

1.4 Your Constant Companions: Our Three Examples 28

1.4.1 moveHUB: The Mobility App for Every Occasion 29

1.4.2 BusinessBooster: The CRM BI Tool for Small and Medium-Sized Enterprises 30

1.4.3 Healthyfy: The Platform for Connecting Patients and Healthcare Professionals 32

2 The Most Important Terms and Their History 35

2.1 The (Pre-)History of Usability and UX 35

2.1.1 Ancient Greece and Hippocrates of Kos 35

2.1.2 The Renaissance and Leonardo da Vinci 36

2.1.3 The Industrial Revolution 36

2.1.4 Work Science 36

2.1.5 Ergonomics 38

2.1.6 Human Factors 38

2.1.7 Software Ergonomics 39

2.1.8 Human-Machine Interaction 39

2.1.9 Human-Computer Interaction 40

2.1.10 Conclusion 40

2.2 Usability and UX Design: The Core Concepts 40

2.2.1 Usability 41

2.2.2 User Experience 42

2.2.3 Design 43

2.2.4 Users 44

2.2.5 Context of Use 49

2.3 The Interplay of Ergonomics, Usability, and UX 50
2.3.1 The Tree Metaphor 50
2.3.2 The Kano Model 54
2.3.3 Ergonomics versus Usability versus UX at moveHUB 56
2.3.4 Ergonomics versus Usability versus UX at BusinessBooster 58
2.3.5 Ergonomics versus Usability versus UX at Healthyfy 60

3 Usability and UX as Key Factors for Outstanding Products 63

3.1 What Is Return on Investment? 64

3.2 Metrics for Business Success 64
3.2.1 Metrics Related to Systems, Products, or Services 65
3.2.2 User- and Customer-Related Measures 66
3.2.3 Service-Related Metrics 67
3.2.4 Choosing the Appropriate Metrics 68

3.3 Planning instead of Discovering ROI 69

3.4 Metrics for Business Success: B2C versus B2B Sectors 71

3.5 What ROI Is Achievable through Usability and UX Design? 73
3.5.1 Studies on the ROI of Usability and UX Design 73
3.5.2 Feedback from Our Customers on the ROI of Usability and UX Design 75

3.6 Case Study: Measuring ROI at moveHUB 76

3.7 Case Study: Measuring ROI at BusinessBooster 77

3.8 Case Study: Measuring ROI at Healthyfy 78

4 How to “Do” Usability and UX Design 81

4.1 The ISO 9241-210 Standard: Human-Centered Design for Interactive Systems 81
4.1.1 Step 1: Plan the Human-Centered Design Process 83
4.1.2 Step 2: Understand and Specify the Context of Use 84
4.1.3 Step 3: Specify the User Requirements 85
4.1.4 Step 4: Producing Design Solutions 86
4.1.5 Step 5: Evaluating the Design 87
4.1.6 Conclusion 88

4.2 Design Thinking 88
4.2.1 Step 1: Understand 90
4.2.2 Step 2: Observe 91
4.2.3 Step 3: Define Point of View 91
4.2.4 Step 4: Ideate 92
4.2.5 Step 5: Prototype 98
4.2.6 Step 6: Test 99
4.2.7 Conclusion 100

4.3 Minimum Viable Product 100
4.3.1 Step 1: Build 102
4.3.2 Step 2: Measure 102
4.3.3 Step 3: Learn 103
4.3.4 Conclusion 104

4.4 Double Diamond 104
4.4.1 Phase 1: Discover 105
4.4.2 Phase 2: Define 106
4.4.3 Phase 3: Develop 106
4.4.4 Phase 4: Deliver 107
4.4.5 Key Principles 107
4.4.6 Conclusion 107

4.5 Usability Engineering for Medical Devices According to the IEC 62366-1 Standard 108
4.5.1 Conclusion 110

4.6 Commonalities among Methods 110
4.6.1 Analysis of Customer Needs 111
4.6.2 Iterative Optimization of the Solution 112
4.6.3 Learning from Direct User/Customer Feedback 113
4.6.4 Speed as an Objective 113
4.6.5 Success through Collaboration 114
4.6.6 Visualization as a Means of Choice 114

5 Data-Driven UX Design 117

5.1 The Goal of the Data-Driven UX Design Process 117

5.2 The Data-Driven UX Design Process 119
5.2.1 How to Get the Most Value from the Process 120
5.2.2 Brief Overview of the Data-Driven UX Design Process 121
5.2.3 Context of Use Analysis, Including Risk Assessment and Documentation 122

5.2.4 Deriving and Documenting Usage Requirements 124
5.2.5 Developing Design Solutions Based on Risk and Usage 126
5.2.6 Evaluation of Design Solutions Prioritized by Risk and Frequency of Use 128

5.3 The Flexibility and Risk Matrix: How Much Data Do You Really Need? 130
5.3.1 The Flexibility of Your Solution 131
5.3.2 The Risk of Your Solution 133
5.3.3 Working with the Flexibility and Risk Matrix 135

5.4 Objective versus Subjective Data 137

5.5 Quantitative versus Qualitative Data 140

5.6 Critically Evaluating Data 141
5.6.1 Where Does the Data Come From? 141
5.6.2 Objective Data or Interpretation 142
5.6.3 Is the Frame of Reference Correct? 143
5.6.4 Have You Had Value Judgments or Statements Explained? 144
5.6.5 Is the Data Complete and Meaningful? 144

5.7 Quick Help for Categorization 145
5.7.1 Do You Need Data during Development? 145
5.7.2 How Much Data Do You Need? 146
5.7.3 What Type of Data Do You Need? 146
5.7.4 Conclusion 147

6 A Helpful Mindset for Usability and UX Design 149

6.1 Assumptions versus Knowledge 150
6.1.1 Clearly Revealing One's Own Assumptions 150
6.1.2 Distinguishing Assumptions from Knowledge 151
6.1.3 Why Do I Need Data for This? 154
6.1.4 Examples of Mostly-False-and-Seldom-True Assumptions 157

6.2 Users Don't Know What They Want! 164
6.2.1 Task and Goal Expertise versus Design Expertise 164
6.2.2 Bringing Together the Expertise 168
6.2.3 Creating Outstanding Solutions with Users 169

6.3 Key Skills for User Researchers: Observe, Ask Questions, and Listen 170
6.3.1 Observing 171
6.3.2 Asking Questions 175
6.3.3 Listening 179
6.3.4 Your Mindset as a User Researcher 181

6.4 Design as a Learnable Process 183
6.4.1 The Role of Creativity 183
6.4.2 The Role of Talent, Practice, Experience, and Knowledge 183
6.4.3 What Do I Need to Learn to Become a Usability and UX Designer? 185

6.5 Testing and Corporate Culture 186

7 Foundations 189

7.1 Psychological Foundations: Don't Worry, It's about People 189

7.2 Perception: The Human Sensory System 190
7.2.1 The Eye 191
7.2.2 The Ear 203
7.2.3 The Skin 211
7.2.4 The Tongue 218
7.2.5 The Nose 220
7.2.6 Conclusion 223

7.3 Human Information Processing 224
7.3.1 Basics of the Model 225
7.3.2 Stimulus 226
7.3.3 Discover 226
7.3.4 Recognize 229
7.3.5 Decide 232
7.3.6 Act 235
7.3.7 Reaction 237
7.3.8 Memory 237
7.3.9 Conclusion 242

7.4 Design Basics 243
7.4.1 Impact of Design 243
7.4.2 Basic Elements of Visual Design 247
7.4.3 Gestalt Principles 269
7.4.4 Interaction Principles 278

8 Data-Driven UX Design in Detail: Preparation 305

8.1 Planning the Data-Driven UX Design Process 306
8.1.1 Who Should Participate in the Stakeholder Workshop? 306
8.1.2 The Agenda of the Stakeholder Workshop 308

8.2 Integrating Data-Driven UX Design into Agile Development Projects 314
8.3 Plan for the Appropriate Amount of Effort 317

9 Data-Driven UX Design in Detail: Context of Use Analysis 321

9.1 What Questions Should the Context of Use Analysis Answer? 322
9.2 What Is the Context of Use Anyway? 324
9.2.1 The Users 324
9.2.2 The Patients (Medical Devices) 325
9.2.3 The Task 327
9.2.4 The Goals 328
9.2.5 The Resources 329
9.2.6 The Environment 331
9.2.7 Risk, Frequency, and Conclusion 333
9.3 Methods of Context of Use Analysis 333
9.3.1 Secondary Research/Desk Research 334
9.3.2 Field Observation 339
9.3.3 Interview/Jobs-to-Be-Done 343
9.3.4 Group Discussions/Focus Groups 350
9.3.5 Quantitative Survey/Questionnaires 355
9.3.6 Contextual Inquiry 360
9.3.7 Analysis of Quantitative Usage Data 364
9.3.8 Web Tracking/Click Analysis 366
9.3.9 Ticket and Feedback Analysis 369
9.4 Analytical Methods of the Context of Use Analysis 373
9.4.1 Task Analysis 373
9.4.2 User or Customer Journey Mapping/Touchpoint Analysis 380
9.4.3 Personas and Empathy Maps 384
9.5 Practical Examples for Context Analysis 391
9.5.1 Context Analysis at moveHUB 391
9.5.2 Context Analysis at BusinessBooster 393
9.5.3 Context Analysis at Healthyfy 397

10 Data-Driven UX Design in Detail: Specifying Usage Requirements 401

10.1 Consider Your Personas or Empathy Maps 402

10.2 Use Cases and User Stories: How Not to Do It and How to Do It Anyway 404

10.2.1 Use Cases 404

10.2.2 User Stories 405

10.3 From User Stories to Requirements 411

10.3.1 Fixed Requirements 411

10.3.2 Range Requirements 412

10.3.3 Requirement versus Wish 416

10.3.4 Documenting and Formulating Requirements 416

10.3.5 Where Do the Values in the Requirements Come From? 418

10.4 Formulating Requirements for moveHUB 420

10.5 Formulating Requirements for BusinessBooster 421

10.6 Formulating Requirements for Healthyfy 422

11 Data-Driven UX Design in Detail: Developing Design Solutions 425

11.1 Reminder: What Is the Goal of Design, and How Do We Go About It? 425

11.2 Design Research 426

11.3 From Coarse to Fine 429

11.4 Developing Basic Solution Options 430

11.4.1 How to Turn a 2 into a 3 432

11.4.2 How to Deal with Too Many 3-Point Ideas 432

11.4.3 Conclusion 433

11.5 Working with Prototypes 433

11.5.1 Horizontal and Vertical Prototypes 434

11.5.2 High- and Low-Fidelity Prototypes 435

11.5.3 Which Prototype Should Be Used and When? 437

11.6 Wireframes 438

11.6.1 Buttons 445

11.6.2 Dropdowns 447

11.6.3 Radio Buttons ... 449
11.6.4 Input Fields or Text Fields ... 451
11.6.5 Combo Boxes ... 454
11.6.6 Checkboxes ... 455
11.6.7 Accordions ... 457
11.6.8 Toggle Buttons ... 458
11.6.9 Sliders ... 460
11.6.10 Calendars ... 461
11.6.11 Time Pickers ... 464
11.6.12 Color Selection ... 466
11.6.13 Information Buttons, Info Buttons, and Tool Tips ... 468
11.6.14 Popups ... 469
11.6.15 Menu Bars ... 472
11.6.16 Carousels ... 474
11.6.17 Webpage Sliders ... 477
11.6.18 Labels and Text Blocks ... 478
11.6.19 Images, Icons, and Placeholders ... 478
11.6.20 Special Case: Warnings ... 480
11.6.21 Conclusion ... 481

11.7 User Interface Quality Check: The Design Grid ... 481

11.8 Screen Flows ... 485

11.9 Click Dummies ... 487
11.9.1 Tools for Creating Click Dummies ... 488
11.9.2 Making the Design Testable ... 491

11.10 Desktop, Mobile, or Smartwatch First? ... 492

11.11 Responsive Design ... 495

11.12 Visual Design ... 497
11.12.1 Visual Design Research ... 497
11.12.2 The Association Space ... 500
11.12.3 Proto-Mood Boards ... 503
11.12.4 Closing Gaps in the Proto-Mood Board ... 504
11.12.5 From Proto-Mood Boards to Mood Boards ... 505
11.12.6 The Visual Design Concept ... 506
11.12.7 Creating Visual Prototypes (Click Dummies) ... 506
11.12.8 The Style Guide ... 507

11.13 Excursion: Low-Code and No-Code ... 508

11.14 Example: Possible Design Steps for moveHUB ... 509

11.15 Example: Possible Design Steps for BusinessBooster ... 511

11.16 Example: Possible Design Steps for Healthyfy ... 512

12 Data-Driven UX Design in Detail: Evaluating Design Solutions 515

12.1 Formative versus Summative Testing 515
12.2 Identifying Knowledge Gaps 519
12.3 User Evaluation versus Expert Evaluation 521
12.4 Remote versus On-Site 522
12.4.1 Dependence on Technical Infrastructure 522
12.4.2 Loss of Important Information 523
12.4.3 Less Control over Confidentiality 523
12.4.4 Non-Digital Products Cannot Be Tested without Bias 524
12.4.5 Conclusion 525
12.5 Designing Test Cases 525
12.5.1 Preconditions 526
12.5.2 Instructions for the User 527
12.5.3 Correct Execution of the Task 528
12.5.4 Postcondition after Correct Execution 529
12.5.5 Determine the Number of Test Cases 529
12.6 The Methods Presented 529
12.7 The Usability and UX Test 530
12.7.1 How Does It Work? 531
12.7.2 When Is It Used? 536
12.7.3 What Do You Need? 537
12.7.4 What Are the Advantages? 538
12.7.5 What Are the Disadvantages? 538
12.7.6 What Are the Alternatives? 539
12.8 Usability and UX Sprint 540
12.8.1 How Does It Work? 540
12.8.2 When Is It Used? 542
12.8.3 What Do You Need? 542
12.8.4 What Are the Advantages? 543
12.8.5 What Are the Disadvantages? 543
12.8.6 What Are the Alternatives? 544
12.9 Rapid User Tests 544
12.9.1 How Does It Work? 545
12.9.2 When Is It Used? 545
12.9.3 What Do You Need? 546
12.9.4 What Are the Advantages? 546

12.9.5 What Are the Disadvantages? 546
12.9.6 What Are the Alternatives? 547

12.10 A/B Testing 548
12.10.1 How Does It Work? 548
12.10.2 When Is It Used? 550
12.10.3 What Do You Need? 550
12.10.4 What Are the Advantages? 551
12.10.5 What Are the Disadvantages? 551
12.10.6 What Are the Alternatives? 552

12.11 UX Benchmarking 552
12.11.1 How Does It Work? 552
12.11.2 When Is It Used? 554
12.11.3 What Do You Need? 554
12.11.4 What Are the Advantages? 556
12.11.5 What Are the Disadvantages? 556
12.11.6 What Are the Alternatives? 556

12.12 5-Second Test 557
12.12.1 How Does It Work? 557
12.12.2 When Is It Used? 558
12.12.3 What Do You Need? 559
12.12.4 What Are the Advantages? 559
12.12.5 What Are the Disadvantages? 560
12.12.6 What Are the Alternatives? 560

12.13 Standardized Questionnaires 561
12.13.1 How Does It Work? 561
12.13.2 When Is It Used? 570
12.13.3 What Do You Need? 571
12.13.4 What Are the Advantages? 571
12.13.5 What Are the Disadvantages? 571
12.13.6 What Are the Alternatives? 572

12.14 Analysis of Quantitative Usage Data 573
12.14.1 How Does It Work? 573
12.14.2 When Is It Used? 573
12.14.3 What Do You Need? 574
12.14.4 What Are the Advantages? 574
12.14.5 What Are the Disadvantages? 575
12.14.6 What Are the Alternatives? 575

12.15 Web Tracking/Click Analysis 576

12.16 Diary Study 577
12.16.1 How Does It Work? 577

12.16.2 When Is It Used? ... 580
12.16.3 What Do You Need? ... 580
12.16.4 What Are the Advantages? ... 581
12.16.5 What Are the Disadvantages? ... 581
12.16.6 What Are the Alternatives? ... 581

12.17 Card Sorting ... 582
12.17.1 How Does It Work? ... 582
12.17.2 When Is It Used? ... 585
12.17.3 What Do You Need? ... 585
12.17.4 What Are the Advantages? ... 586
12.17.5 What Are the Disadvantages? ... 586
12.17.6 What Are the Alternatives? ... 586

12.18 Expert Review ... 587
12.18.1 How Does It Work? ... 587
12.18.2 When Is It Used? ... 589
12.18.3 What Do You Need? ... 589
12.18.4 What Are the Advantages? ... 589
12.18.5 What Are the Disadvantages? ... 590
12.18.6 What Are the Alternatives? ... 590

12.19 Association Space Test ... 591
12.19.1 How Does It Work? ... 591
12.19.2 When Is It Used? ... 592
12.19.3 What Do You Need? ... 593
12.19.4 What Are the Advantages? ... 593
12.19.5 What Are the Disadvantages? ... 593
12.19.6 What Are the Alternatives? ... 593

12.20 Eye Tracking ... 594
12.20.1 How Does It Work? ... 594
12.20.2 When Is It Used? ... 597
12.20.3 What Do You Need? ... 597
12.20.4 What Are the Advantages? ... 597
12.20.5 What Are the Disadvantages? ... 598
12.20.6 What Are the Alternatives? ... 598

12.21 Physiological Measurement Methods ... 599
12.21.1 How Does It Work? ... 600
12.21.2 When Is It Used? ... 600
12.21.3 What Do You Need? ... 601
12.21.4 What Are the Advantages? ... 601
12.21.5 What Are the Disadvantages? ... 601
12.21.6 What Are the Alternatives? ... 601

12.22 Analyzing Errors Using Root Cause Analysis 602

12.23 Learning from the Data and Making Informed Decisions 608

12.23.1 Implementing Feedback 608

12.23.2 Documenting Decisions 610

12.24 Evaluations in Medical Device Development 610

12.25 Example: Evaluation Methods for moveHUB 611

12.26 Example: Evaluation Methods for BusinessBooster 612

12.27 Example: Evaluation Methods for Healthyfy 614

13 Design, Get Data, Repeat: When Is It Enough? 617

13.1 Determining the Transition Based on Quality 618

13.2 Determining the Transition Based on Time 619

13.3 Should You Readjust? The Flexibility and Risk Matrix as an Aid 620

13.4 Choosing the Right Approach 622

13.5 What's Next for moveHUB 623

13.6 What's Next for BusinessBooster 623

13.7 What's Next for Healthyfy 624

14 Decision Matrix for Method Selection 625

15 Wrap-Up and Conclusion 633

15.1 It's All about the Users 633

15.2 Usability and UX Design Is Interdisciplinary 634

15.3 UX Is Not Designed, It Is Experienced 634

15.4 The Path to an Outstanding Product Is Iterative 634

15.5 Once You Start with Usability and UX Design, You Never Stop 635

15.6 Design Based on Data 635

15.7 Closing Words and Call for Participation 636

Appendices 633

A **Glossary** ... 639

B **Bibliography** ... 653

C **The Authors** ... 663

Index ... 665

Service Pages ... I

Legal Notes ... II

Foreword

The more widespread technical devices become in everyday life, the more varied the demands placed on interactions with these devices. In the case of professional applications, the focus is often on error-free functionality, optimized service life, and user acceptance, especially by improving task completion. In a private context, using technical devices often should be fun.

Fulfilling the requirements placed on usability and user experience (UX) to a high degree is becoming an essential prerequisite for the economic success of technical products. To achieve this high degree of fulfillment, a corresponding design process is necessary that considers human requirements in every design phase and with a high degree of traceability. Using extensive data on human usage behavior is a promising approach for improving human-centric product design.

This handbook offers a comprehensive and well-founded compilation of processes, methods, and applications for improving the usability and UX of technical products. In particular, data-driven UX design (3DUX) is a design approach with great potential for future product designs. In the interest of optimized human-machine interaction, I hope this handbook has a wide distribution.

Prof. Dr. Ralph Bruder
President of the Carl von Ossietzky University of Oldenburg, Germany

Chapter 1
Introduction

"Compared with the ability to arrange the work of a single day in a meaningful way, everything else in life is child's play."
—Johann Wolfgang von Goethe (poet and naturalist)

Dear reader, good to have you here! In this first chapter of our book, you'll learn who we wrote it for, why this book is a good investment of your time, and how to make this book work best for you. We'll also introduce you to three examples that we'll return to again and again throughout the book. We'll use these examples to help you better understand the methods and approaches presented in each chapter. In the following chapters, we assume that you have read the introduction to these examples, that you're fundamentally familiar with the products described, and that you know by whom and how they are used. So, it's worth reading this chapter first, even if you want to be a bit more selective in your reading afterward. However, before we get to our recommendation for how to read this book, we would like to start with exploring whether this book is suitable for you.

1.1 Who Is This Book Written For?

During the conception of this handbook, we thought long and hard about who the audience for this book is and who would benefit the most from it. Especially when you're already familiar with a topic like user experience (UX) design, deciding who needs what content isn't an easy task. Much knowledge seems self-evident because of years of experience and daily involvement with a topic. Through our work in our data-driven user experience (also known as 3DUX) design agency, *Custom Interactions,* and our teaching activities at the Technical University of Darmstadt, we keep coming across the most diverse "levels" of expertise—from complete beginners to seasoned professionals. We have consciously decided not to write this book exclusively for experts but also to offer inexperienced readers a guide and introduction to the wonderful world of usability and UX.

So, we have you in mind if you belong to one of the following groups:

- You're studying and would like to learn something about the practical application in addition to your studies' theoretical content and concepts.

- You're new to your profession and dealing with usability and UX design for the first time.
- You're already involved in product development and are responsible for creating or further developing a product. In this book, we'll cover business-to-consumer (B2C) products (products sold directly to end customers), business-to-business (B2B) products (products sold to companies), and medical devices.
- You're already working in the area of usability and UX and would like to add to your repertoire of methods or would like assistance in selecting the appropriate method.

But of course, we are also happy if you have a different background and are still interested in our handbook. This broad claim about the book's readership is also reflected in the objectives and structure of our handbook. Let's look at the objectives of this book next.

1.2 Why Should You Read This Book?

You're holding the *Usability and User Experience Design* handbook in your hands. Both terms are widely used, and many product managers and students have at least heard of them. In daily practice, however, *usability* and *UX* have not yet arrived in every company and every product development, which has increasingly far-reaching consequences since users and customers are becoming more demanding every day, especially in these areas.

Over the past 10 to 15 years, this change has been evident in B2B applications developed for enterprise customers. A few years ago, a technical function was sufficient for product use or purchase. Today, ease of use and intuitive use often come first—even if this requirement is not always explicitly expressed.

In the B2C sector, that is, for products developed for end users, this trend began much earlier and is now much more advanced. In this context, good usability is now a prerequisite, and the focus is on designing a comprehensive UX—a trend that can also be observed increasingly in B2B applications and will play a significant role in shaping the future success of B2B products.

We already hear the call for better usability and UX every day in our usability and UX labs. So, unless your product technically does something completely new, unprecedented, or until now unattainable (like flying to Mars), you need strong usability and a strong UX for an outstandingly successful product. So, you should read this guide if one of your goals is precisely that—a successful product.

We have divided the book into two parts help achieve this goal. We'll cover the basics in the first part of the handbook (Chapter 1 through Chapter 7). This part includes a general understanding of the terms and origins of the discipline, as well as an overview of various approaches and processes that have been established over the years. We also want to show you a mindset that we have found conducive to success in our work as usability and UX professionals. The basics then conclude with an overview of the essential perceptual psychology and design concepts to prepare you soundly for designing your products. The book's second part then goes through the stages of the data-driven UX design process step-by-step, showing you how to use the fundamentals you've learned to design outstanding products.

We hope to offer a comprehensive overview of usability and UX design and to enable you to acquire knowledge in this subject area, explore contexts, and learn new methods. We have committed to a balancing act: On one hand, we want to impart systematic knowledge. On the other hand, we want to focus on practice, which is especially important to us as practitioners. After all, what is the point of purely theoretical knowledge that is difficult to transfer regarding practical issues?

For an overview of the subject area, we would like to draw a map, figuratively speaking, to help classify topics and think about them. For each point on the map—each topic in this book—other literature is of course available that you can refer to understand individual issues, methods, and concepts in more detail.

Wherever we have suitable recommendations for you, we'll refer to them in the relevant section of the book. So don't worry: Even if you know your way around, you'll surely get one or two further tips.

So, after reading this book, you should not only have an overall view of usability and UX but should also be able to decide in a concrete, practical situation whether an issue is a usability or a UX issue and whether you need to look elsewhere for a suitable solution.

For this purpose, we'll provide you with a decision-making tool that you can use to locate your question in the project. With this knowledge, you can choose from the pool of methods presented to find the right and most appropriate approach.

After all, we want to support you in getting the best possible results for the users of your products. We want to help you develop the best version of your product: intuitive, with exactly the suitable range of functions, easy to use, inspiring, and, last but not least, successful.

For this purpose, we have structured the book in the following way:

- **Chapter 2: The Most Important Terms and Their History**
  After we have (hopefully) helped you with this first chapter to determine if our handbook is helpful for you and how you can best work with it, we start directly with the definition of terms in Chapter 2. With this chapter, we want to prepare some common ground so you can easily follow this book and also exchange ideas with other practitioners.
- **Chapter 3: Usability and UX as Key Factors for Outstanding Products**
  In Chapter 3, we'll show you why considering usability and UX when developing

your products is worthwhile. For this purpose, we have dug deep into the literature, will explain studies, but also talk a bit out of the box and present measurable results from our customers.

- **Chapter 4: How to "Do" Usability and UX Design**
  Chapter 4 is devoted to an overview of various product development processes that aim for high usability and an excellent UX. This overview should help you navigate the process maze and select suitable processes and steps.
- **Chapter 5: Data-Driven UX Design**
  Chapter 5 introduces the *data-driven UX design (3DUX)* process. We have derived this process from existing processes but with a stronger focus on critical and complex products that consider risk in use. The 3DUX process has proven itself in our daily work and helps combine the best of the various process worlds. As the name suggests, we follow a data-driven approach that propagates the integration of user data and user feedback in as many places as possible and aims to guarantee a successful product.
- **Chapter 6: A Helpful Mindset for Usability and UX Design**
  Chapter 6 deals with a—from our point of view—wholly underestimated but fundamentally important topic: the appropriate mindset for working in the field of usability and UX. In this chapter, we point out various logic traps and mental blocks that we, and many of our colleagues, have fallen into repeatedly over the years. But of course, we don't stop with describing these mental blocks but want to help you develop and strengthen a mindset we have proven to be successful and helpful in our work in usability and UX.
- **Chapter 7: Foundations**
  Chapter 7 then touches on some technical basics. Usability and UX design form a discipline that requires knowledge from various fields. We'll introduce you to the psychological basics of perception, memory, and information processing, as well as the fundamentals of design and the laws of design.
- **Chapter 8: Data-Driven UX Design in Detail: Preparation**
  Starting in Chapter 8, we'll look at the data-driven UX design approach together in detail—always using our three examples, of course, for the greatest possible practical relevance.

  Chapter 8 is about preparing and planning for the data-driven UX design process. We address practice-relevant questions, including how to proceed in agile development processes and how you can estimate the development effort.
- **Chapter 9: Data-Driven UX Design in Detail: Context of Use Analysis**
  Then, in Chapter 9, we look at the context of use analysis. You'll learn what the context of use is, why it should not be neglected when considering usability and UX, what belongs to it, and which methods you can use to get to know usability and UX design best in practice.

- **Chapter 10: Data-Driven UX Design in Detail: Specifying Usage Requirements**
  In the following Chapter 10, we'll introduce you to usage requirements and show you how to formulate them so that, in the end, you create a product that delights your users. We'll also look at communicating usage requirements to internal and external stakeholders, such as technical development partners.
- **Chapter 11: Data-Driven UX Design in Detail: Developing Design Solutions**
  Building on usage requirements, we devote ourselves to the design solutions in Chapter 11. In this chapter, we'll introduce you to the different variants of prototypes and provide you with various methods and tools that you can use directly for your designs.
- **Chapter 12: Data-Driven UX Design in Detail: Evaluating Design Solutions**
  In Chapter 12, we look at the final step of the data-driven UX design process: evaluating design solutions. In this chapter, we provide an overview of several methods. Additionally, we support you in choosing the proper methods to answer the right questions.
- **Chapter 13: Design, Get Data, Repeat: When Is It Enough?**
  In Chapter 13, we address a question that many of our customers ponder: When does it end? Especially in an agile, iterative process, this decision can be challenging. In this chapter, we provide some concrete decision-making aids.
- **Chapter 14: Decision Matrix for Method Selection**
  Chapter 14 supports your daily work when you have finished reading this book. We provide a decision matrix of the methods presented in this book and guides you, through simple rules, on making design decisions.
- **Chapter 15: Wrap-Up and Conclusion**
  We conclude this book in Chapter 15 with a summary and conclusion.

In this book's appendices, you'll find the bibliography and a glossary containing this book's most important definitions and terms for reference.

Feedback Welcome!

If you have read the book and think important content is missing, we'll happily receive feedback! Feel free to get in touch at *handbook3DUX@custom-interactions.com*. We welcome your suggestions for the next edition of this book.

1.3 How Should You Read This Book?

Now that this book is in your hands, as should already be clear from the page count—we've got a lot to say about usability and UX. This book is intentionally broad and comprehensive to combine an overview of the subject and the basics from different disciplines with valuable, practical information. We understand if you ask yourself at this point: Is a book with more than 600 pages still *usable*? Therefore, let's consider a few ideas about how to use this book first.

We have written the *Usability and User Experience Design* handbook so you can read it from cover to cover without any problems. We introduce concepts once in Chapter 2 and later assume you know their relevant definitions later. However, you're not forced to read every chapter from cover to cover. Therefore, we refer to the basics when necessary or summarize them. We believe this approach will facilitate access to information even if you read from front to back, as it will save you from scrolling to a definition introduced four chapters ago.

However, we also wrote the book for readers with varying levels of prior experience, and we didn't assume that you, as a perhaps experienced UXer, will read it all the way through. Feel free to jump to the parts most relevant to you. In addition to dividing the book into a basic section and a practical section, we have done the following to achieve this flexibility:

- We reference introductory chapters (Chapter 2 through Chapter 7) and still briefly summarize the essential points. Without reading the basics, you can understand the contents or read up on them in detail.
- We have added a glossary to the book that contains the most important definitions so you can look up unknown terms previously defined in other chapters quickly.
- We have included all methods in a decision matrix in Chapter 14, with references, so you can jump directly to a method of interest and focus only on that method.

So, use the book as needed: as interesting literature that you browse through from front to back, as a source of inspiration that you use selectively, or as a reference guide that you take off the shelf repeatedly. If we get you to read this book not just once but to consult it often, we'll have achieved more than we had hoped for.

1.4 Your Constant Companions: Our Three Examples

To strengthen the practical relevance of this handbook, we would like to introduce our three case studies on the following pages. An upfront disclaimer: These examples are fictitious, and any similarities to living or deceased products, companies, and persons are coincidental.

Before we had these three examples, we thought long and hard about the best example to take you by the hand and make our content clear. However, as the readership of this handbook grew, so did the challenge of finding a single example that would be relevant and transferable to you and everyone else. We could not find that *one* example.

Instead, we decided to develop three different examples. Each of these examples has a different focus and struggles with specific challenges. We chose these examples to represent as wide a range of applications and problems as possible. Of course, even with these three examples, covering all conceivable combinations is impossible. Nevertheless, we hope you can identify with at least one of the three examples and find parallels to your product.

To make reading more natural, we have additionally equipped these examples with a past, a present, and a future. In this way, we want to simulate everyday work in product development, which rarely takes place on a blank slate, without worries. So, our protagonists always deal with questions that our customers ask themselves, and they face the same challenges and decisions. Throughout the chapters, our protagonists apply the methods and tools presented in each case and learn as they go—just as they would in actual product development teams. But enough of the introduction, let's get to know our three products and the teams behind them.

1.4.1 moveHUB: The Mobility App for Every Occasion

First, we would like to introduce you to the fictitious *moveHUB*, which is a B2C startup from a rural area near Austin, the capital of Texas, located in the center of the state. The three founders of moveHUB—Matthew, Emma, and Anthony—met while studying in Austin. All three share the same tale of woe: they come from more rural areas outside of Austin and could not afford an apartment directly in Austin due to the cost. Instead, like so many other students, they commuted throughout their studies. Although their homes are only about 15 miles from Austin, poorly connected public transportation and confusing alternatives lead to an hour of travel time for each commute.

When their university announced an ideas competition, moveHUB was born. Matthew, Emma, and Anthony decided to work on an idea that makes commuting between rural regions and cities easier and gives users a transparent overview of the available means of transportation.

Due to the planned type of use—namely, on the road—the three founders decided that moveHUB should be launched as an app downloadable via the app stores of the relevant providers. Since travel interests everyone, they didn't want to focus on a single smartphone operating system, such as Apple's iOS, but rather develop a solution everyone could use. However, because their resources as a startup are minimal, Matthew, Emma, and Anthony decided to start with the release for Android first since the user numbers in this segment are disproportionately larger.

Beyond these primary considerations, however, the team still has many questions and few final decisions. No app exists yet, nor any business model or formal company at this point. All we have is a quick logo sketch, shown in Figure 1.1, and big dreams. In our book, we'll accompany Emma, Anthony, and Matthew on their journey.

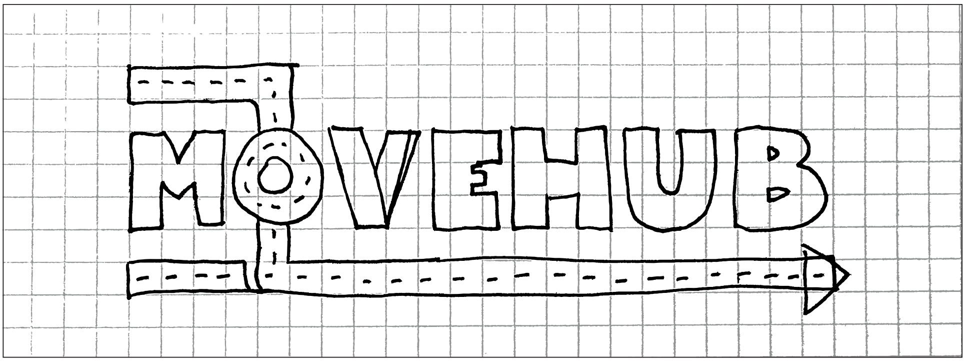


Figure 1.1 First logo idea for the fictitious startup moveHUB.

1.4.2 BusinessBooster: The CRM BI Tool for Small and Medium-Sized Enterprises

Our second fictitious example throughout this book comes from SME IT Inc., which is a medium-sized B2C company with 64 employees in Hamburg, Germany. The company was founded in 1984 by Markus Meier directly after his studies in electrical engineering.

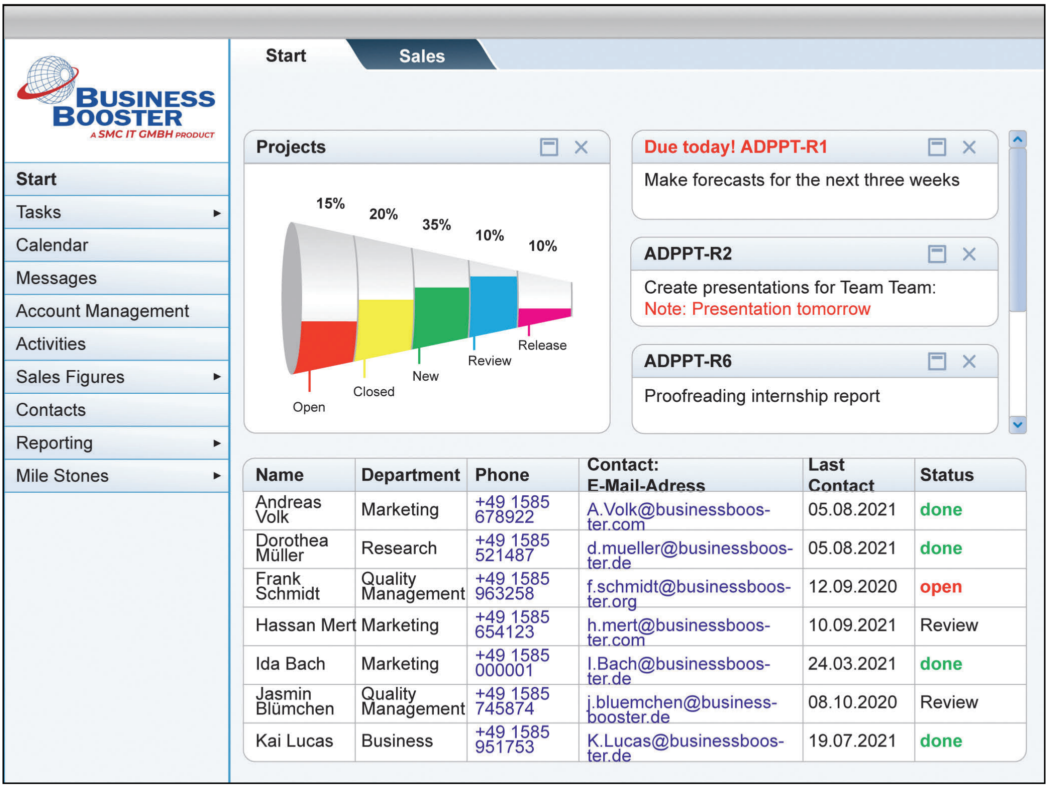
SME IT Inc. aims to provide small and medium-sized enterprises (SMEs) with IT solutions to make their lives easier and make their businesses more successful.

The company's bread-and-butter product is the software BusinessBooster, as shown in Figure 1.2, which Markus developed at the time. BusinessBooster is software enabling small and medium-sized companies to manage their contacts and track sales activities. In addition, the software also offers the ability to create quotes and invoices and record payments received. Thus, the product represents a mixture of *customer relationship management (CRM)* software and an invoicing and accounting tool.

BUSINESS
BOOSTER
A SME IT Inc. PRODUCT

Figure 1.2 SME IT Inc.'s BusinessBooster logo.

In the early days of the software, BusinessBooster could set itself massively apart from its competition. The novel functions and the link between the two worlds were unique, and the clear presentation of figures and evaluations made the product a bestseller, as shown in Figure 1.3. This precise data presentation proved so valuable that SME IT Inc. could gain significant market share with its solution within a few years and ultimately become the market leader.



Due today! ADPPT-R1

Make forecasts for the next three weeks

ADPPT-R2

Create presentations for Team Team:
Note: Presentation tomorrow

ADPPT-R6

Proofreading internship report

Name	Department	Phone	Contact: E-Mail-Adress	Last Contact	Status
Andreas Volk	Marketing	+49 1585 678922	A.Volk@businessbooster.com	05.08.2021	done
Dorothea Müller	Research	+49 1585 521487	d.mueller@businessbooster.de	05.08.2021	done
Frank Schmidt	Quality Management	+49 1585 963258	f.schmidt@businessbooster.org	12.09.2020	open
Hassan Mert	Marketing	+49 1585 654123	h.mert@businessbooster.com	10.09.2021	Review
Ida Bach	Marketing	+49 1585 000001	I.Bach@businessbooster.de	24.03.2021	done
Jasmin Blümchen	Quality Management	+49 1585 745874	j.bluemchen@business-booster.de	08.10.2020	Review
Kai Lucas	Business	+49 1585 951753	K.Lucas@businessbooster.de	19.07.2021	done

Figure 1.3 BusinessBooster's user interface (UI).

Due to this great success, the SME IT Inc. team grew to more than 110 employees in the meantime and was able to launch additional, complementary IT products for SMEs. Although all subsequent products were also quite successful, to this day, BusinessBooster is by far the top-selling product and accounts for almost 70% of SME IT Inc.'s total sales.

Although SME IT Inc. is constantly working on further developing the software, the product is visibly getting on in years. Sales figures have been declining for almost ten years now. More and more complaints and customer inquiries are coming in. Many existing customers have switched to the competition in the meantime—even though the effort to switch is enormous for customers. Market leadership has become a distant prospect for Markus Meier and his team. As managing director, Markus has even been forced throughout the last few years to give notice to some employees in service and development or to not fill new positions due to lack of funds.

Markus is also finding it increasingly difficult to attract new, younger, and modern employees to his company. He knows from feedback in job interviews that the seemingly outdated product is also contributing to the problem.

After much hesitation, Markus decides to part with the current version of the BusinessBooster software and launch a completely new software as BusinessBooster 2.0. This software will be reprogrammed from scratch and fix all the problems that have been scaring away potential customers.

Markus has only technical solutions and additional features in mind. He also looks forward to implementing the product in a modern programming language and is sure that this choice will provide real added value for his clientele.

If Markus is honest, he admits he also fears the new development. Above all, he is afraid of losing his remaining loyal existing customers. Thus, he also wants to approach the redesign somewhat cautiously. This approach also fits well with the economically strained situation in which SME IT Inc. finds itself. The company really has only one shot to modernize the software and must use this chance to win new customers.

Now, you've met Markus, SME IT Inc., and BusinessBooster, and we'll accompany him in later chapters.

1.4.3 Healthyfy: The Platform for Connecting Patients and Healthcare Professionals

As our final fictitious example, we would like to introduce Healthyfy, which is a B2B and B2C mediation portal for health experts. As a spin-off of a large medical technology company, it intends to facilitate the leap into the purely digital world. The parent company of Healthyfy, known as MedTec Inc, is in device development and manufactures monitoring devices and ventilators for intensive care units. To diversify its portfolio, the medical technology company, MedTec Inc, plans to develop an exclusively digital healthcare offering.

Healthyfy aims to create a digital platform where doctors from different specialties and patients can log on via a web interface. Patients should be able to search for a suitable specialist and make appointments online. Doctors should be able to accept and reject patients, keep appointments, and communicate with patients. Communication should function in various ways. A chat function and a video function are being considered—but no one knows yet. The dream behind Healthyfy is to map a doctor's visit entirely digitally, which will likely make Healthyfy a medical product.

Currently, the Healthyfy team consists of 11 people who have been given a fixed budget for 18 months by the parent company. During this time, the first version of the portal should be developed. Those responsible at the parent company are aiming for a *proof of concept* to prove that further investment in this area is worthwhile.

MedTec Inc plans to spin off Healthyfy formally if this succeeds. In this case, Linda Miller—the current team leader of the Healthyfy team—would presumably take over the management and lead Healthyfy into a hopefully glorious future. Linda is relatively new to the company and was hired specifically for the Healthyfy project. Before joining MedTec Inc, Linda worked for a startup in the software industry, which she successfully accompanied to exit.

We meet Linda and Healthyfy at a time when, besides the team, only the strategy document; a logo designed by the parent company's marketing team, as shown in Figure 1.4; and generous funding is in place. It's an ideal starting position, even if Linda is a bit nervous about the highly regulated environment and the many procedures in medical technology. We'll see what Linda and her team make of Healthyfy.

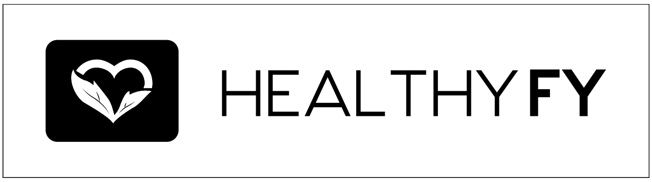


Figure 1.4 The logo for Healthyfy.

Chapter 2

The Most Important Terms and Their History

"The beginning of wisdom is the definition of terms."
—Socrates (philosopher)

We want to create a strong common foundation for the exchange of information and start this chapter by summarizing important terms, their meanings, and their relationships. In this way, you'll become familiar with the technical terms related to usability and user experience (UX) design, and subsequently, you can use them correctly in practice.

We start with a look into the past to introduce you to the origins of the field. This history lesson will help you understand why different manifestations and implementations of usability and UX design exist. Then, we look at the current state of the relevant technical terms you'll need for your daily work as a usability and UX specialist.

2.1 The (Pre-)History of Usability and UX

Both terms from the book title, *usability* and *UX design*, are the result (and still part) of an evolution of different disciplines that have dealt with people-related issues. For us, the historical development of our field is exciting and vital for this handbook's completeness and can help you classify different approaches in practice and understand focal points. If you're not interested in history, you can safely skip this section.

In compiling this section, we focused less on exact historical developments and more on the logical derivation of the various content orientations in the discipline of usability and UX design—colored, of course, by our own professional background. A different selection is conceivable, of course. Other experts would perhaps present other milestones to you at this point; we, however, will tell you about the historical development from our perspective.

2.1.1 Ancient Greece and Hippocrates of Kos

Around 400 BC, the Greek physician Hippocrates recognized that the practice of certain crafts caused specific health damage (Draxler & Cheung, 2010). Furthermore, in his surgical treatises, he wrote detailed instructions for the layout of operating rooms, the arrangement of instruments, lighting, and the duties of the people present (Adams, 1868). The recognition of the relationship between work and injury, on the one hand, and the specification of work procedures, on the other, are still fundamental to the concepts of *ergonomics* and *work science*, which have had a formative influence on the development of usability and UX design.

2.1.2 The Renaissance and Leonardo da Vinci

Leonardo da Vinci contributed other foundations to today's view. The Renaissance polymath is known for his works of art and many designs for machines and buildings. To adapt devices to human beings, he intensively studied their movement capabilities. He wrote down the resulting physiological and biomechanical findings as laws (Schlick, Bruder, & Luczak, 2010).

At this time, again, the essential elements of ergonomics and work science concepts can be found, which were transferred to the general public to a greater extent during the Industrial Revolution.

2.1.3 The Industrial Revolution

Now, let's jump to the Industrial Revolution in the second half of the 18th century. Following Fürstenberg (1981), cheap and unskilled labor was massively available before the Industrial Revolution. Thus, from an economic perspective, concerns about working on a large scale with people, with people's work, or even with people's well-being were nonexistent.

Only when divisions of labor emerged during the Industrial Revolution did the need for planning and consistent quality increase. Also, with increasing mechanization and mass production and the accompanying clocking of work, a new form of pressure to perform arose. As a result, work design became the focus of interest. Scientific disciplines emerged that dealt with optimizing labor and labor activity and human performance capabilities and limitations—first and foremost, work science and ergonomics.

2.1.4 Work Science

The concept of work science, including the first definition, dates back to 1857. As you'll see, the term ergonomics also originates from the same article and the associated first definition.

The definition given in this instance essentially revolves around the *humanization* and *rationalization of* the world of work.

Humanization of the World of Work

According to the *Gabler Wirtschaftslexikon* (German business dictionary), the humanization of work means measures aimed at improving the content of work and working conditions to make the world of work as humane as possible (Bartscher, 2021).

Rationalization of the World of Work

In its most common usage, rationalization describes those measures that serve to increase productivity and profitability, which ultimately results in a restriction to the value target (Voigt, 2021).

Humanization versus Rationalization of the World of Work

Looking at these two terms, you can quickly see that humanization and rationalization are challenging to achieve simultaneously. Especially under difficult working conditions, where the availability of labor was not the limiting factor, an increase in efficiency has often led to terrible working conditions in the past. However, today, the thinking is entirely different. The objective is to increase productivity while improving conditions for the people in the system. The basis for this approach can be found in the science of work.

Work Science as a Discipline

This approach is precisely where *work science* comes in. The goal of work science is to keep an eye on both aspects. Due to the importance of human resources, work science promotes rationalization oriented towards humanization goals (Schlick, Bruder, & Luczak, 2010, p. 7). In simple terms, the science of work tries to make work more pleasant and less damaging for people while at the same time helping companies make better use of their human workforce.

According to Luczak and Volpert (1987), various goals are involved, such as the following:

- Creating activities that are harmless, feasible, tolerable, and free of impairment
- Meeting standards in terms of social appropriateness of work content, work task, and work environment, as well as remuneration and cooperation
- Making room for maneuvering, acquiring skills, and developing one's personality

All these goals can be achieved through systematic analysis, which involves the ordering and design of the technical, organizational, and social conditions of work processes.

The focus of work science is on the work process itself. The main focus is optimizing physical processes while considering anthropometric characteristics (e.g., body size or strength). Ultimately, the question is how to construct and set up a workplace in the best possible way to achieve the two goals—no harm to people and the highest potential productivity.

Even though work science procedures and methods could be used purposefully in the private, non-work-related sector, this transfer has largely been denied due to the discipline's focus—not so with ergonomics.

2.1.5 Ergonomics

The terms *work science* and *ergonomics* are closely linked and are often mentioned in the same breath. Depending on the interpretation, ergonomics is regarded as a subarea of work science—or as going beyond work science. The first joint appearance of the two terms in the mentioned article from 1857 also points to proximity in terms of content.

According to the International Ergonomics Association (2021):

> *"Ergonomics is the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data, and methods to design in order to optimise human well-being and overall system performance." (International Ergonomics Association)*

Even if the literal derivation of the word *ergonomics* from the Greek *ergon* (work) and *nomos* (law) indicates a reference to work, the definition of the international umbrella organization clarifies its system-related view, which coincides with the second meaning of *ergon* (completed work). Ergonomics is, therefore, not limited to the sector of work.

2.1.6 Human Factors

The term *human factors* is also broader in scope and does not only refer to the world of work. The term covers various factors that influence people in socio-technical systems. The US military coined this term in the 1940s (Tossell et al., 2016). At that time, the aim was to reduce human errors, for example, when piloting aircraft (Fitts & Jones, 1947).

In contrast to the terms *ergonomics* and *work science* introduced in the previous sections, however, *human factors* are less about people's physical and anthropometric or bodily characteristics and more about psychological and cognitive traits. Social aspects also play a decisive role in the "human factor." Ergonomics is thus closer to work science, while human factors are related to psychology.

As technical systems become more complex, new questions are continually being raised. For example, one of the more recent questions is about the cooperation between a human and a machine to solve a task jointly. A concrete example is assisted driving of an automobile with adaptive cruise control. At the so-called "first level," the vehicle automatically accelerates and brakes depending on the distance to the car or truck in front (SAE International, 2018). However, the driver remains in charge of steering the vehicle, so that the task of "driving the vehicle" is performed jointly by man and machine.

As just described in the case of human factors, technological changes also lead to new or changed issues in the other disciplines mentioned earlier. In the case of software ergonomics, a new subdiscipline has even emerged, which we'll cover in the next section.

2.1.7 Software Ergonomics

With the advent of PCs, classic ergonomics has evolved into software ergonomics. This linguistic adaptation can be seen as an attempt by the scientific discipline not to lose relevance in a world with an increasingly digital focus and to modernize itself (our own interpretation). Following the Gabler definition, software ergonomics aims to design software products according to the people working with them (Gabler Lexica). The contents of the software ergonomics are, for example, questions of the dialog organization, work contents, and the preservation and organization of decision areas.

Following the word's root, software ergonomics can also be located in the world of work. This orientation was essential and logical in the first years after PCs emerged. However, the intense work focus has replaced the term, as PCs, software, and other digital media have increasingly penetrated the work environment into the private domain. In the meantime, the term *human-machine interaction* is more commonly used across the board, which we'll explain shortly. The term *human-computer interaction* has become established for the subarea of software ergonomics, which we'll cover subsequently.

2.1.8 Human-Machine Interaction

Human-machine interaction originates in the analysis, design, and optimization of physical machines. We understand human-machine interaction as any exchange of information between humans and machines. This information can trigger actions (e.g., operations, human responses) but does not have to. We also understand information conveying status information between the human and the machine without a direct request for action as human-machine interaction.

Our definition is deliberately rather broad and includes all technical systems that exchange information with people. These systems can involve the operation of passenger aircraft in the cockpit, or a punch press activated with a classic switch. On the one hand, human input; on the other hand, machine output via one or more user interfaces (UIs) are made. In each case, the goal is to ensure the smoothest possible collaboration. The discipline of designing optimized human-machine interactions thus already has the purpose of optimized usability in mind, even if only for a subset of products.

2.1.9 Human-Computer Interaction

The field of human-computer interaction pursues similar goals to human-machine interaction but focuses on interaction with computers and other digital systems, that is, less on the physical aspects and more on the mental and cognitive factors. This field is thus primarily about mental performance and its limitations and impairments. Of course, the physical design of products also plays a role since PCs or smartphones are often gateways to digital applications. But the focus of action is often on the digital aspects.

According to Hamborg (2021, translated from German), "human-computer interaction [...] deals with the study, design, and evaluation of human-computer (hardware and software) interaction."

So, if you were looking for a discipline that researches and develops the relationship between humans and computers, you would most likely find success under the term human-computer interaction. Human-computer interaction, software ergonomics, and human-machine interaction, however, all strive to achieve good usability and UX.

2.1.10 Conclusion

All the disciplines we've mentioned so far are concerned with technology design, albeit with different goals. On one hand, our goals are about productivity and error prevention; on the other hand, about people, their performance, and their needs.

Technology should be designed to fit people, not the other way around—fortunately, historical development is heading in precisely this direction. We would like to see technology fade into the background and provide the best possible support to help people achieve their personal and professional tasks and goals. The focus should be about the "what" and not the "how." This vision personally drives us and motivates us again and again. So much so that we founded the data-driven UX design (3DUX) agency *Custom Interactions* in 2013 based on this basic idea. We include these goals on our website as part of our corporate vision.

Two other concepts are central to achieving this vision and are also fundamental to the disciplines mentioned in this section: *usability* and *UX*. Let's turn to these core concepts next.

2.2 Usability and UX Design: The Core Concepts

Now is the time to look at the core concepts of this book—namely, *usability* and *UX*—before we then look at the umbrella term *design*. Later in this section, we'll provide more details about additional core terms, such as "users."

2.2.1 Usability

The definition of usability as we know and use it today dates back to the 1970s. However, the first mentions of usability can be found earlier in the context of the design of weapon systems. Today's definition has been modified to show that besides (military) products, systems and services also have usability. Since 1997, International Organization for Standardization (ISO) 9241-11 has been a standard that defines the term "usability" uniformly worldwide. Since the revision of the standard in 2018, usability has been defined in the following way:

> *"Extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." (ISO 9241-11:2018)*

Looking closely at this definition, notice how it uses the terms "specified user," "specified context of use," "specified goals," and "effectiveness, efficiency, and satisfaction." We'll address these four terms directly in this section. Because of the central importance of the *user* as well as the *context of use* for this book, we have devoted separate sections for each term (Section 2.2.4 and Section 2.2.5).

Effectiveness, Efficiency, and the Goal

The term *effectiveness* describes the accuracy and completeness with which a goal can be achieved. *Efficiency* describes the extent to which resources are used in relation to the results achieved. In defining usability, one question is whether a user can do what they want and how much effort is required for their goal. To assess both effectiveness and efficiency, therefore, the *goal* itself must be clear.

A simple example: If my goal is to travel to Munich from Los Angeles, then effectiveness indicates whether I arrived in Munich. I was not particularly effective if I was stranded in New York instead. Efficiency, however, measures my use of resources, such as time. If I travel from Los Angeles to Munich by the most direct route possible, I am significantly more efficient than going from Los Angeles to Sydney and then to Munich.

In this example, effectiveness and efficiency are measured objectively. Measuring effectiveness is quite simple because it is a yes/no decision. If you want to know whether a system, product, or service has high effectiveness, then you can ask the following questions:

- Was the task solved correctly? (Yes/No)
- How many errors occurred during task completion when the result was incorrect? (This question measures the extent to which a single task can be solved effectively.)
- What is the proportion of completely solved tasks out of the total tasks for a user?
- What is the ratio between users who could successfully solve the task and the users who failed to do so?

Measuring efficiency is more complex and always involves considering the effort invested from the users' perspective. Effort can be measured in time, money, energy, or other costs (e.g., assistance required). Consequently, significantly more possible parameters exist for efficiency, such as the following:

- The time needed to solve the task
- Number and severity of errors in the performance of the task despite correct result (as a measure of invested mental effort)
- Number of supports required, such as manuals, instructions, or other advice (as a measure of the intellectual effort invested or, if applicable as in the case of paid support, even of the financial effort)

Satisfaction

The third measure in the definition of usability according to ISO 9241-11:2018 is *satisfaction*, which is defined in the following way:

> *"Extent to which the user's physical, cognitive, and emotional responses that result from the use of a system, product, or service meet the user's needs and expectations." (ISO 9241-11:2018)*

Put simply: Did the solution meet the user's expectations?

Satisfaction is much more subjective than effectiveness and efficiency and is less easy to measure. Therefore, a variety of different approaches are available to measure the satisfaction of your users. Most commonly, scale questions are used ("How satisfied were you with …?"). However, other approaches can measure satisfaction using open-ended questions or physiological data (e.g., by analyzing facial expressions).

Focus on the Use

The three usability measures—effectiveness, efficiency, and satisfaction—are designed to evaluate a system, product, or service. You cannot measure effectiveness and efficiency outside of use, only while a user is interacting with a system, product, or service. Furthermore, to measure satisfaction, usage must have already occurred. Therefore, satisfaction only be assessed retrospectively to the actual use. The usability measurement is thus always linked to the actual use of the system, product, or service.

2.2.2 User Experience

The term *user experience* also goes back to the 1970s. It was used at that time, for example, in the title of the publication "User Experience With the CYBER Graphics Terminal" (1974) by E. C. Edwards and David J. Kasik. However, the term did not become widespread until Don Norman, one of the great thought leaders in human-computer interaction, assigned himself the new job title "User Experience Architect" at Apple Computer in 1993 (cited in Knemeyer & Svoboda, 2022).

The term UX considers the idea that factors outside the direct use of a system, product, or service also influence its evaluation. The ISO 9241-11:2018 standard defines UX in the following way:

> *"User's perceptions and responses that result from the use and/or anticipated use of a system, product, or service." (ISO 9241-11:2018)*

The definition of UX focuses on a user's preferences, attitudes, and emotions as well as physical and psychological reactions (including perceptions of trust, safety, security, and privacy). This definition explicitly includes not only actual use but also expected use. UX thus takes a holistic view of people's perceptions. According to Hassenzahl (2010, p. 2), UX is a special case of human experience in general. Therefore, the system, product, or service differentiates between a general experience and a UX.

Due to the holistic view and the focus on the experience, UX is often initially not located in the working world but purely in the private sphere. This imbalance may sound understandable initially, but in practice, many work systems, products, and services are now falling short of their potential. At this point, we would like to point out that this approach is too simplistic: In the working world, too, people want a great UX for the systems, products, and services they use.

The professional discussion in science has also struggled for a long time with what should be considered part of usability. Some pioneers researched *joy of use* around 2000 and tried to establish a more holistic view. In 2002, for example, Burmester, Hassenzahl, and Koller pointed out the importance of joy, emotions, and fun in the use and design of interactive products.

The greatest challenge in UX lies in its design itself. While precise units of measurement can be used for optimizing usability, and the solutions are often much easier to find, designing a targeted UX is much more difficult. Targeting depends on understanding the basic needs and personal tasks of the users. Since these aspects are often different in kind or different in degree, the individually experienced UX is also different. For example, one measure of UX could be a user's increased attachment to a system, product, or service.

Note that—even though we write it this way—we don't design the UX as in "the experience itself." Instead, we design the system, the product, or the service that subsequently meets the human being and triggers the experience we hope for.

In practice, design—and thus all its subdisciplines—is often equated with "making things beautiful." However, this motto only covers a fraction of what can be achieved with usability and UX design. But what do we mean by *design*?

2.2.3 Design

The term *design*, or the specification *usability and UX design*, comes more from practice and less from science. However, since we are writing a book for practitioners and not a theoretical textbook, we decided to use *design* as a generic term. To us, when we talk about design, you should have a broad meaning of the word in mind. We don't just mean "making things beautiful," but also design in the sense of "development."

For us, usability and UX design is an activity that deals with designing systems, products, and services to achieve high usability and an excellent UX. To clarify this aspect, we recommend always using the word "design" in connection with the objective (in this case, usability and UX design).

So, when we talk about "usability and UX design" in this book, we mean, without exception, an iterative and human-centered process that puts users and other stakeholders at the center of interest at all times. We describe such a process in detail in Chapter 4 and in Chapter 5. In the next section, we'll define what we mean by *users* and *stakeholders*.

Usability and UX design form a discipline that requires a variety of expertise. The classic design degree is only marker of expertise—an important one, of course, but by no means sufficient to work human-centered alone. Especially in UI design, other disciplines are often successfully represented (e.g., psychology or humanities). Holistic usability and UX design also require *user research*, which provides valuable input for product, system, and service design. User research has the central task of answering what needs to be implemented and whether the chosen implementation is promising.

Nevertheless, to put it bluntly, you'll rarely find all the required competencies in a single person. Instead, various specialists usually work together to find the best solutions. But let's now move on to the main players—people who use the product.

2.2.4 Users

When we talk about human-centered design and data-driven UX design, the essential term—and probably the one you'll read most often, along with *usability* and *UX*—is the *user*.

ISO 9241-11:2018 defines the term *user* in the following way:

> *"Person who interacts with a system, product, or service." (ISO 9241-11:2018)*

This definition is really about the person who uses the system, product, or service themselves and/or is in direct contact with it.

Note that the standard adds that users are not only individuals who directly use the system, product, or service but also individuals who perform the following activities:

- Operate the system, product, or service
- Use the results of the system, product, or service
- Support the operations (including maintenance and training) of the system, product, or service

Therefore, the group of users can become quite large, although we would always recommend that you only consider users whose actions also influence the design.

For example, if you're developing a simple web application, you don't have to worry about the overnight office cleaners. These individuals may come into contact with the computer during the cleaning process and thus come into contact with your product, but they do not directly use the web application itself (or its results) in any way.

However, if you're developing a patient monitor for an intensive care unit in a hospital, you should certainly think about "cleaning and disinfection." In this case, too, your product will not be used directly by cleaning staff, but accidental adjustment of operating elements could have far-reaching consequences.

From the two examples, you can see how discussions about who belong as users are important in practice. In Chapter 4 and Chapter 5, we take a risk-based approach to this decision.

User/User Group

In addition to the term *user,* often *user groups* are relevant for usability and UX design. We've noticed that using the term *user* exclusively often gives the impression that it's about a single person. In part, that's true: You observe individuals and get feedback from individuals, but you don't design for individuals; you typically design for groups of users.

A user group is defined in ISO 9241-11:2018 in the following way:

> *"Subset of intended users who are differentiated from other intended users by characteristics of the users, tasks, or environments that could influence usability." (ISO 9241-11:2018)*

Therefore, a user group is a group of people who use the product, system, or service and who are similar to each other in terms of specific characteristics, the tasks performed, and the environment being used. Working with user groups is always essential when different groups of users must be considered to achieve an appropriate usability and UX design.

Thus, to help you decide early on whether you have different user groups, a crucial task is to identify the relevant characteristics, tasks, and environmental conditions that differentiate each group. For this goal, ask yourself the following questions:

- Must users have specific *characteristics* to complete the task successfully?
- Do all the people you want to consider together have the same *tasks* with the solution?
- Do the *environmental conditions* differ between these users in a way that impacts the usability or UX when using my solution?

You may often have your own feelings about a certain kind of experience. However, a best practice is to question your preconceived notions again based on the three criteria: characteristics, tasks, and environmental conditions. In practice, we've found that often users are split into too few (or too many) user groups.

In addition, you might hear the term "target group" instead of "user group." The term *target group* originates from the marketing sector. Therefore, the presentation of a target group usually contains different information, as this concept is intended primarily for addressing buyers in advertising. This information differs from the information needed to design a suitable product, system, or service. Consequently, we'll stick to the term *user group*.

Stakeholder

In addition to users, the term *stakeholder* is often used in discussions about new developments or further development. ISO 9241-11:2018 contains a definition of "stakeholder" that is just as broad and vague:

> *"Person or organization that can affect, be affected by, or perceive themselves to be affected by a decision or activity." (ISO 9241-11:2018)*

In other words, a stakeholder is anyone who feels they are somehow involved or affected.

The fact that the term is so broad makes determining who should be considered a stakeholder accurately difficult. However, you can also consider people who would not otherwise be classified as users because of their tasks.

A typical example in a professional environment could be the decision-makers in an organization. These individuals are not users of a system, product, or service but are still interested in its functionality or its outcomes.

In a private environment, let's say you want to buy a hot tub for your family and place it in the backyard. In this case, your neighbors are stakeholders because they can't use the hot tub, but they do hear and see your family daily. Therefore, they may have wishes and expectations for the successful implementation of your hot tub.

Who you should consider as a stakeholder always depends on your system, product, service, and objective in the development. Society as a whole could be a stakeholder, depending on how broadly you define the term; for example, there are startups and companies that rely heavily on sustainability as an argument in their value propositions.

Customer/Buyer

In many environments, the stakeholder most often mentioned is the *customer* or *buyer*, who makes the purchasing decision and buys the product or is considering buying the product.

In the business-to-customer (B2C) environment, these individuals are usually the same people as the users. However, in some exceptional cases, users and customers are different people entirely, for example purchasing gifts and products for babies and small children or buying products for people in need of care.

In the business-to-business (B2B) environment, you'll almost always encounter a separation between buyers and users. Buyers can come from purchasing, management, or another department, or a buying can be the legal entity. Users are usually the people who use the system, product, or service in the name and interest of the company.

As in the B2C sector, exceptions exist in the B2B sector as well. For example, if a CEO buys an online course to improve leadership skills, this purchase is made by the company for a user within the company. Nevertheless, the decision-maker, the buyer, and the user coincide in the same way as is usually the case with B2C systems, products, or services. For marketing purposes, you would proceed differently than in the typical B2B environment, where the buyer and the user are different people. We call this area "B2C-like" and choose an approach adapted to it in our projects (in, for example, Chapter 3, Section 3.4).

Let's now return from the rare cases to the more general. In a nutshell, our understanding of a customer or buyer is as the person who makes a purchasing decision and spends the money, regardless of whether or not they subsequently use the product. Therefore, the group of customers/buyers can be the same as the group of users, but being the same is not required.

The difference between *customer experience* and *UX* arises from the fact that these groups of people (buyers versus users) do not have to be the same people. The customer experience is concerned with the comprehensive purchasing experience of a person, whereas the UX is concerned with the comprehensive experiences of users.

Imagine again you're buying a hot tub for the backyard. Since a hot tub is a rather expensive product, you would probably want relatively high-quality information, good advice, a smooth purchase process, and prompt delivery. In addition, you expect excellent service in case problems arise and always expect a competent contact person. Of course, you also want the hot tub to work well and for you to have fun using it. But the use itself is only a tiny part of your customer experience. However, if you happen to have children, then your children—depending on their age—might judge the hot tub solely on whether it is fun to swim in or whether rubber duckies are allowed. Your UX would then be in contrast to the UX of your children.

So, in this example, the UX would be part of the customer experience because you bought the hot tub and use it yourself. Now, let's say you bought a wading pool instead of a hot tub, and the wading pool is too small for you to use: Then, you would have a customer experience, but not necessarily a UX, except that you still need to inflate, clean, repair, and store the wading pool.

Note that the customer experience can encompass much more than just the UX. The UX can be part of the customer experience, but it doesn't have to be. For that reason alone, dealing with both aspects makes sense. This book focuses on usability and UX and only, in rare cases, on customer experience. However, especially in the early stages of product development, the benefits you gain from your work will be more significant if you look not only at usage but also at the purchase and the reasons for purchase.

User versus Customer Using Our Three Examples

To distinguish further between users and customers, let's turn to our three examples and determine who is a user and who is a customer. Let's start with the *user*:

- For moveHUB, the user is the person searching for transportation using the app.
- In BusinessBooster, the user is the person who sits at the computer and creates and manages contacts and checks sales figures. Depending on the task area, you might need several user groups, each of which only processes a portion of the tasks.
- Different user groups exist for the Healthyfy platform: On the one hand, we have doctors who post their profiles on the health platform and make appointments, and on the other hand, we have the patients who search for a suitable doctor on the portal.

Thus, for each of our examples, we have at least one user group, which makes sense since our solutions must be operated or used to fulfill their purposes.

In addition to the users, however, a customer buys or could buy the product. In this context, too, let's look at our three examples:

- The customer of moveHUB is the person who pays for the use of the app or the means of transport. Advertisers would also be possible customers. The customer in this context depends on the business model—how the company decides to make money.
- The customer, in the case of BusinessBooster, is the company that purchases the software. The ultimate decision-maker in the company (i.e., the person who signs the contract) could be the managing director, the purchasing department, or the department manager.
- At Healthyfy, different customers are at play: the doctors who want to present themselves on the platform, the patients looking for a doctor, and advertisers who wish to address doctors or patients.

Conclusion

We hope we've clarified several points through our examples in this section, such as the following:

- The user and the customer can be identical (as in the moveHUB example), but they don't have to be.

- Users are always real people. Customers can also be companies, which real people, of course, represent. However, contracts are concluded with the company.
- You can recognize a user by whether they are trying to accomplish a specific task with your solution.
- You can recognize a customer by the money they give you.
- A change in product features can change who uses your product because, for example, certain tasks become (un)solvable. However, changing tasks that can be solved with the product also influences the customer's purchase decision.
- Changing your business model can change who buys your solution.

2.2.5 Context of Use

In the definition of usability from the ISO 9241-11:2018 standard, the term *context of use* is used. This concept is central to human-centered design, as you'll see later in this section. First, let's look at the standard, where the context of use is defined in the following way:

> *"Combination of users, goals and tasks, resources, and environment."*
> *(ISO 9241-11:2018)*

A context of use, therefore, describes the specific combination of a user group that performs a precisely outlined task with clear goals in a specific environment with defined resources. The environment in this sense can be technical, physical, social, cultural, or organizational. Note, however, that changing just one of these parameters can enormously impact usability.

Imagine, for example, in your private life, you find it amusing that your smartphone plays a funny sound when you receive a message. If we put you in an important professional meeting with the same smartphone, you would probably be embarrassed by the same sound. In this example, we have the same user (you), product (smartphone), and task (information when a new message arrives). However, the environment, the user's role, and with that, the context of use have changed. In our example, this change affects the satisfaction of the user.

This simple example shows the central importance of the *context of use* for usability and UX design. If you do not know the context of use, you cannot achieve excellent usability and UX. This concept also illustrates how no *one* usability of a system, a product, or a service exists. Usability always depends on the *context of use* and, thus, on the user and the specific situation.

The context of use analysis is, therefore, the first step in the data-driven UX design process (see Chapter 5). Due to its central importance, we have also dedicated Chapter 9 to the context of use analysis.

Like the user groups described in Section 2.2.4, contexts of use also differ in terms of user characteristics, tasks, and environments. In the first case, however, the focus is on describing the characteristics of *users,* whereas the description of the context of use focuses on the *environment*. So, it mainly changes the perspective of the description. As in our earlier example with the smartphone and the funny message sounds, the context of use is about a specific combination. The user group, in turn, is defined more broadly to include all users who perform the same tasks in the same environments and have the same user characteristics. You could also create a user group summarizes all persons with the same contexts of use for a system, product, or service. Or, put another way, you could have two user groups differ that based on their different contexts of use.

2.3 The Interplay of Ergonomics, Usability, and UX

At this point, we would like to show the three terms *ergonomics*, *usability*, and *UX* in interaction to support you in determining the focus of your activities.

Even though the terms are regularly used synonymously, their targeted use can help you clarify the primary goal you want to achieve with your work. To illustrate this point, we would like to introduce you to the tree metaphor.

2.3.1 The Tree Metaphor

If you were to imagine the design of systems, products, or services as a tree, our understanding is that ergonomics would be the roots of a tree, as shown in Figure 2.1.

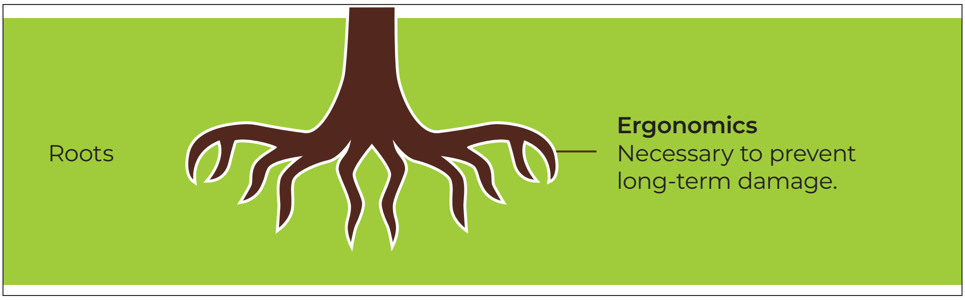

Roots
Ergonomics
Necessary to prevent long-term damage.

Figure 2.1 In our tree metaphor, ergonomics is the root of a system, product, or service.

Without roots, nothing happens, and if the roots are no good, problems arise in the long run that can't be intercepted either. Thus, we appeal to you to always think about ergonomics (even if the topic may be less "sexy" than UX). You'll realize ergonomics is essential when you try to work for a day with some cutting-edge VR goggles in a sophisticated application. The application's usability may be good; maybe even the UX of the application (considered in a short period) is fantastic. But when your neck kills you in the evening, and you can barely move in pain because the glasses were too heavy or the cable was too short, and you had to adopt a forced posture, you'll realize quickly why ergonomics is so important.

However, few plants consist only of roots, and so is the case here. To maintain the metaphor, the usability of a solution can be equated with the trunk, as shown in Figure 2.2.

The trunk is the solid base needed to design a good product. As with the definition of usability described in Section 2.2.1, a user has a specific goal that should be achieved as effectively, efficiently, and satisfactorily as possible. The product, system, or service is redundant if this goal is impossible. In our tree metaphor, usability thus forms the trunk that supports the rest of the tree.

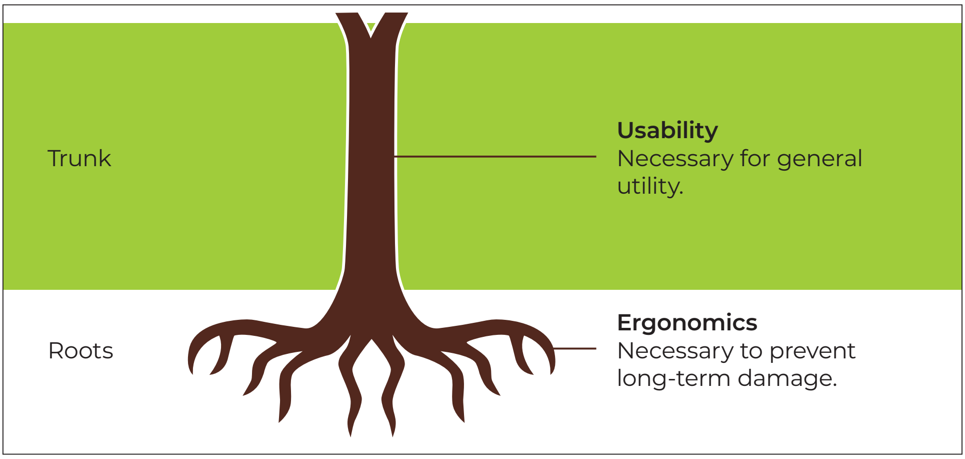

Trunk
Usability
Necessary for general utility.
Roots
Ergonomics
Necessary to prevent long-term damage.

Figure 2.2 Usability, in our tree metaphor, is the trunk of a system, product, or service.

For example, imagine that you buy a new smartphone. This smartphone looks fantastic. With a new type of casing, if you were asked, you would say that it is the most beautiful smartphone you have ever seen. It also excites you that it feels great in your hand (positive UX). It is neither too heavy nor too light and does not slip away (thoughtful ergonomics). Unfortunately, you cannot insert a SIM card into the smartphone, and an eSIM is not supported. You can, therefore, neither make phone calls nor access the Internet on the go (effectiveness is not provided). You can connect to a WLAN at home, but the connection is established slowly, as with a modem. Before a new web page has built up, you must wait a minute with your new smartphone (its efficiency is poor). What do you think: How excited would you be about the smartphone now? We suspect that you would not be enthusiastic and that your initially positive attitude towards the smartphone would quickly turn negative (negative UX due to usability not being provided).

In our example, we have already talked about enthusiasm, and now, we turn to the realm of UX, which in our metaphor, forms the crown of our tree, as shown in Figure 2.3.

What needs to happen for you to be excited about a system, product, or service? Maybe you're helped to feel competent, or you're enabled to learn something new, which stimulates you. Or maybe it allows you to show who you're or makes you feel connected to others. Or maybe the product is very appealingly designed, and you like it a lot.

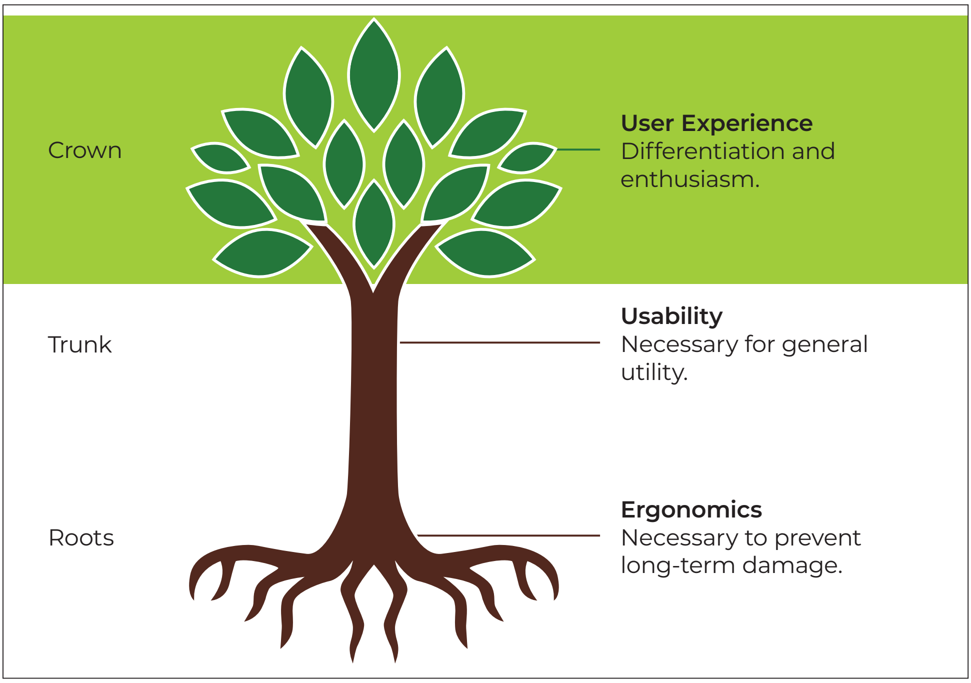

Crown
Trunk
Roots
User Experience
Differentiation and enthusiasm.
Usability
Necessary for general utility.
Ergonomics
Necessary to prevent long-term damage.

Figure 2.3 In our tree metaphor, UX is the crown of a system, product, or service.

You may have noticed how often we've used the word "maybe"—that's because no *single* ingredient will make any system, product, or service "wow" you.

To generate enthusiasm, the *needs of the users* must be addressed. These needs are manifold and also still vary from person to person. In Sheldon et al. (2001), a whole list of human needs is provided. Hassenzahl and Diefenbach have developed short needs lists in various works over several years. In our view, the catchiest presentation of their abbreviated list can be found in Diefenbach et al. (2014). To summarize those needs relevant in the context of technology use and thus contribute to a good UX, consider the following:

- **Competence**
  The need to face and overcome challenges. The experience of self-efficacy and success plays a role here.
- **Connectedness**
  The need to feel close to others, especially those important to you. It's about the feeling of social inclusion and closeness.

- **Meaningfulness**
  The need to capture meaningfulness and gain new insights.
- **Stimulation**
  The need to learn about new things. Curiosity, entertainment, or distraction often play a role here.
- **Security**
  The need to be able to plan things and be safe from threat and uncertainty. It's about a sense of relaxation through planning and structure.
- **Popularity**
  The need to be recognized by others, to be someone others find exciting or emulate. Fame and responsibility play a role here, as do power and influence.
- **Autonomy**
  The desire to decide things freely. Self-determination, freedom, and independence play a role here.

Suppose you integrate features into your system, product, or service that address the needs of your user groups. In that case, you can differentiate your solution from competitors and support user loyalty. If you exceed the expectations of your user groups, then your system, product, or service will inspire them.

Looking again at the image of the tree, as shown in Figure 2.4, all aspects that contribute to a fantastic UX represent the tree's crown.

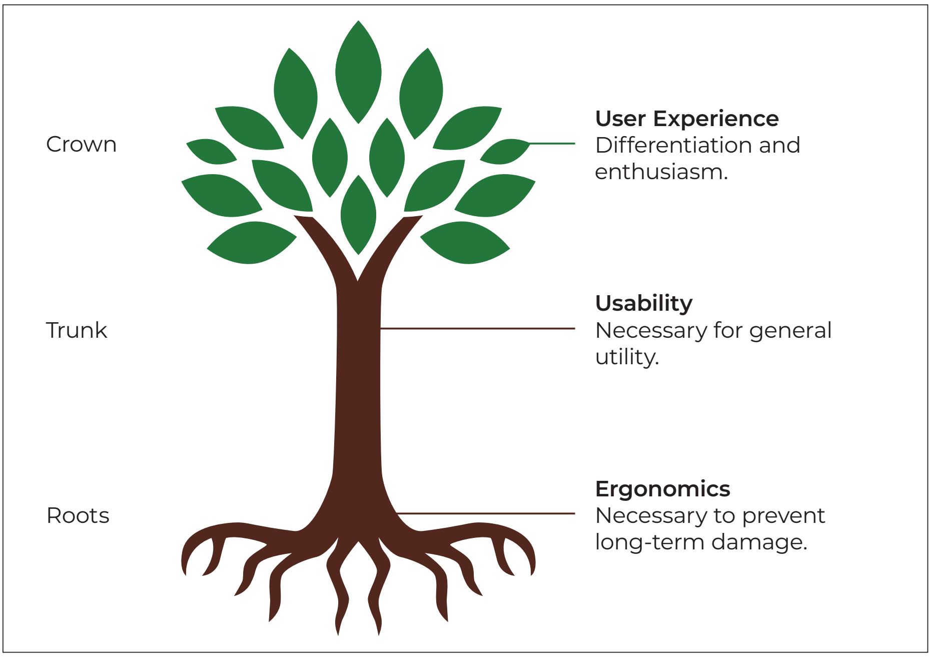

Crown
Trunk
Roots
User Experience
Differentiation and enthusiasm.
Usability
Necessary for general utility.
Ergonomics
Necessary to prevent long-term damage.

Figure 2.4 The complete tree with ergonomics as the root, usability as the trunk, and UX as the crown.

This picture is not 100% accurate because UX also includes the perception of usability and ergonomics. Recall our earlier smartphone example: In this case, poor usability contributes to a poor UX. For the tree crown to stand in full splendor—that is, for a system, product, or service to have an excellent UX—it needs a solid trunk and healthy roots. But the crown is the eye-catcher and gets the most attention.

In this chapter, we would like to discuss another aspect that we've already indirectly alluded to. We wrote earlier that you could inspire your user groups by exceeding their expectations. But these expectations change over time. A product feature that inspires enthusiasm today may be expected next year. This progression over time can be explained with the relatively well-known Kano model, which we present in the next section.

2.3.2 The Kano Model

The Kano model is a model for describing customer satisfaction. Compared to other approaches, this model allows you to understand how satisfaction with a product changes over time and what this change depends on. In this model, established by Noriaki Kano and colleagues (1984), the first two dimensions are *customer satisfaction* and *(customer) expectations*, as shown in Figure 2.5.

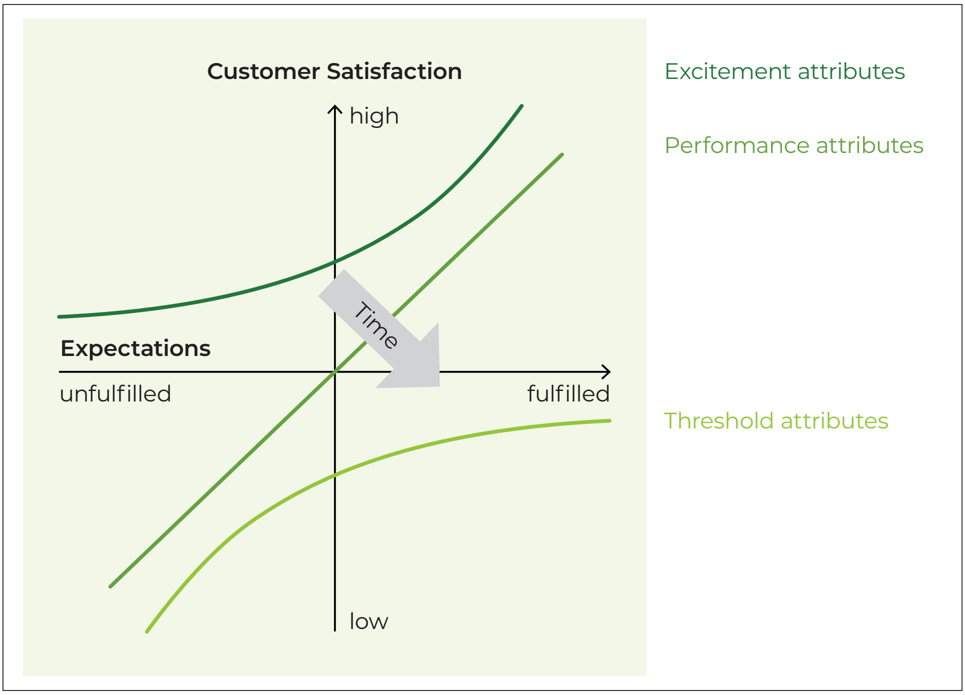

Customer Satisfaction
high
low
Expectations
unfulfilled
fulfilled
Time
Excitement attributes
Performance attributes
Threshold attributes

Figure 2.5 The Kano model of customer satisfaction.

The better expectations are met, the higher customer satisfaction. However, the Kano model divides the characteristics of a system, product, or service into four categories:

- Threshold attributes (also called must-be or basic)
- Performance attributes (also called one-dimensional)
- Excitement attributes (also called attractive or delightful)
- Rejection attributes (also called reverse)

Threshold attributes include features that are expected but not explicitly formulated by the user. A user might be unaware of these features, and only when not present does their necessity become apparent and customer satisfaction drops. In our smartphone example, the feature “make phone calls on the move” would be such a threshold attribute: When you buy a new smartphone, you assume you can make phone calls without explicitly paying attention to this feature. However, if this feature is absent after the purchase, your satisfaction drops directly. On the other hand, fulfilling this feature does not trigger any enthusiasm in you either: The feature is expected.

Regarding our tree metaphor, threshold attributes are often found within ergonomics (root) and usability (trunk).

If some features of ergonomics or usability are core aspects of your system, product, or service, they can also fall into the category of *performance attributes*. Users are explicitly aware of these features and actively use them to compare different solutions. The more performance features are fulfilled by your system, product, or service compared to the competition; the more likely people are to buy from you and the higher their level of satisfaction.

In our smartphone example, an outstandingly good camera could be important for potential customers. They then subject every potential smartphone candidate to a critical quality check concerning the camera and decide on the model that appeals to them best in this category (combined with all other performance attributes). However, although all the threshold and performance attributes might be *fulfilled*, that does not mean they are *thrilled*.

According to the Kano model, you need *excitement attributes* to increase customer satisfaction significantly. These features are rated as relevant by the user and exceed their expectations. In our tree metaphor, we are in the crown area. Typically, a single or just a few enthusiasm features are enough to trigger enthusiasm. In our smartphone example, you were excited by the look and how the device fits in your hand. Sometimes, that’s enough.

The fourth feature category involves *rejection attributes*, which is the opposite of the other three types. If rejection attributes are fulfilled, then customer satisfaction drops. Like performance characteristics, users are explicitly aware of these characteristics and try to avoid them. For example, if you don’t like a particular manufacturer’s brand, then in our smartphone example, a rejection feature might be if the device came from that manufacturer. You would have to "wrestle" with whether to buy the device anyway.

The Time Factor

Time plays a crucial role in evaluating products, systems, and services, as the expectations of customers and users adapt over time (thus to changing experiences and conditions). The importance of time is illustrated by the gray arrow in the Kano model shown in Figure 2.5. Over time, excitement attributes become performance attributes and, finally, threshold attributes.

Let's take the smartphone as an example again, but this time looking specifically at the presentation of the iPhone in 2007, when its large display and operation via multi-touch triggered veritable storms of enthusiasm. These features clearly differentiated the iPhone from other cell phones and appealed strongly to the target group's needs. If you look at this feature today, the fact that you operate a smartphone with a large multi-touch screen has become a threshold attribute in most cases.

Thus, be aware that you must constantly evolve your system, product, or service today to continue to delight users tomorrow.

Conclusion

Based on our experience, we are firmly convinced that ergonomics, usability, and UX are inseparable. From our point of view, the field is in flux, with many different influences, expertise, sciences, and backgrounds come into play. Perhaps, we have overlooked some topics that you would have given more weight to. If you have any additions, please send us an e-mail to *handbook3DUX@custom-interactions.com*.

Now, let's look at how these terms might differ in practice through our three examples.

2.3.3 Ergonomics versus Usability versus UX at moveHUB

First, let's consider moveHUB, our B2C mobility app for every occasion. Our three founders are highly motivated and want to start with something. Preferably, the three of them want to start programming right away. Anthony—the best programmer on the team—has already come up with a backend solution to store the data in the best possible way. Given the current world Anthony is in, his questions might be "How do we even get public transportation data? Do the local transit associations have APIs we could dock on?"

Emma is a business engineer. Although she also programs quite well as a hobby and is often Anthony's sparring partner, she is responsible for the financing. Anthony, however, quickly inspires Emma to reconsider her position because, in his words, "If we can't get the technology right, we can forget everything." The team soon abandons the "financing" role in favor of another programmer role.

Work goes on for a while, and the backend is making progress. The first data sets can be accessed, and after endless hours of pondering, the proper database and the appropriate programming platform have also been found. Emma and Anthony are proud and happy.

Only Matthew has reservations. He thinks it's incredible what the team has achieved but doubts that it should continue in the same way. Perhaps this unease is because he doesn't know anything about programming, and the other two are diving into worlds that make him feel like he's just slowing them down. So, Matthew seeks out startup literature. He reads a lot, and the idea begins to mature that technology is not the only component of a successful product.

Matthew gathers all his arguments and brings the other two from the programming world to a shared table. This meeting has an effect. The team takes a programming break to think about other important issues. Although they didn't know it then, they have taken a step that most startups miss.

In his reading, Matthew found a tree metaphor that he would like to apply to his product. The three of them start working on it in a workshop.

In the end, the team concludes that ergonomics is an issue they have little influence over. After all, they are designing an app for a smartphone, and thus the hardware is fixed. They also believe the app is not used for an exceptionally long time either, so there should be no signs of fatigue from using it. Nevertheless, the three also come to the topic of one-handed operation and the arrangement of buttons. Their idea: When traveling, you often have luggage in your hand, so the app should be conveniently operable with one hand. So, they think about how long a typical thumb is and in which areas essential controls should be placed. They also touch on the issue of supporting both left-handed and right-handed users, as shown in Figure 2.6.

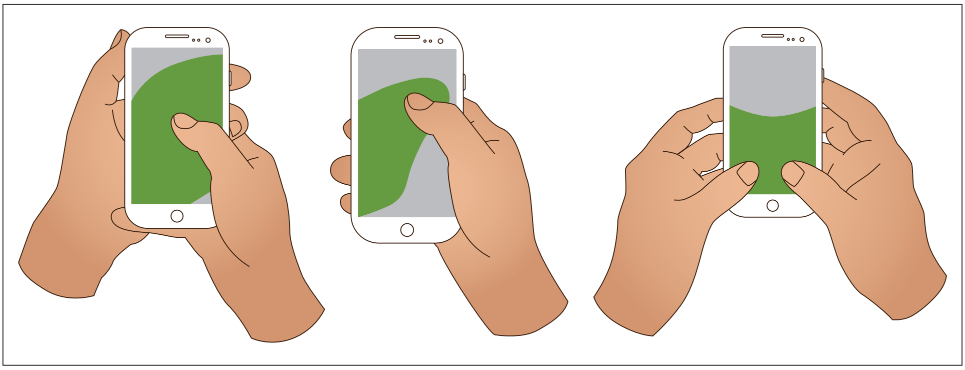


Figure 2.6 Operating a smartphone using one or two hands. The green areas are the areas that are easiest to reach.

Their bottom line is that they want to make sure that the most important controls are in an area that can be safely operated one-handed by both left-handed and right-handed users.

In terms of usability, efficiency and effectiveness immediately stand out to the founders. They had already decided that the app would only be used briefly. Therefore, the app should provide important information quickly. Ideally, the app would open on the right screen and deliver the information users seek. They decide the time until the desired information is found should be optimized.

On the one hand, the information presented is a characteristic of effectiveness and the possibility to impress the users. What if the product has a kind of intelligence? What if it knows where people want to go and doesn't just display any information about departure times but instead shows only the relevant ones? This approach could increase effectiveness as well as efficiency, but above all, this feature could also delight users. Anthony, Emma, and Matthew decide to think about how to achieve this feature later.

2.3.4 Ergonomics versus Usability versus UX at BusinessBooster

What about BusinessBooster, our B2B customer relationship management (CRM) business intelligence (BI) tool for small and medium-sized enterprises? Markus Meier had already decided that the modernization should focus primarily on technical features. Above all, he expects changing to a modern programming language to bring tremendous added value for customers, making it easier and quicker to respond to customer requests for new functions and making it possible to call up the data from different computers via the web interface.

He decides that this information needs to be spread. After all, customers keep dropping out, and the remaining ones now must stay on board. Markus can use the news of the programming language change to lift customers' spirits. He happily picks up the phone and calls his best customer. Getting put through to the managing director, Markus announces the development of BusinessBooster 2.0 to fundamentally modernize the software.

The reaction is exactly what Markus had hoped for. His customer is thrilled and admits he has already looked around for other providers. But now he is willing to wait. Two questions are burning in the customer's mind, however: When can he expect the new version, and what will change? Those are the questions Markus has been waiting for. He enthusiastically presents his plan to implement BusinessBooster 2.0 in a modern programming language. However, this response does answer the customer's question, and the customer still wants to know what will change. Irritated, Markus repeats that the new modern programming language is the central added value. The customer suddenly gets quiet. He then says clearly and directly, "What's the point of a modern programming language for me?"

That hits. Markus finishes the conversation and then must take a moment. The reaction was unexpected: He failed to convince his best customer of the added value. So, he picks up the phone again. But the results stay the same. Calls to other customers also fail to produce the desired result.

Markus decides to gear the realignment more closely to customer requirements. Together with his team, he scrutinizes the use of BusinessBooster. In ergonomics, they arrive at the idea that BusinessBooster is software that is probably used throughout the entire workday. It mostly runs on desktop computers or laptops—although the team assumes these are connected to larger monitors. All in all, it assumes a desktop environment with a mouse and keyboard. However, the team also notes that truly little is known about the work conducted with BusinessBooster. They don't know the typical user and therefore find it difficult to judge the ergonomics. As a first measure, they decide to talk to the top 10 customers to find out more.

The same realization is also forming when it comes to usability. The team believes it knows quite a bit about this topic. But it also realizes that usability is quite uncertain. In principle, all the findings are based on "It's always been done that way" and "It's always worked that way." For Markus, however, the ten telephone calls with his customers were such traumatic experiences that he no longer can accept this situation. Obviously, the company doesn't know its own customers because otherwise, the redevelopment of the solution would work—that's his new motto.

At a high level, the team understands that it must support the user's tasks in the best possible way. Effectiveness and efficiency must be high, which is especially true for B2B software. Markus has since learned this lesson, and he was embarrassed that he had always dismissed usability as "making things pretty." The team now also identifies processes that have been completed in the earlier software and makes an initial assessment. But it also realizes that you often go around in circles and end up in a battle of opinions.

The situation is no better in the area of UX. How could you inspire a customer? The team has various ideas—but the uncertainty is suddenly enormous. Before the phone calls, they were sure they knew what would excite their customers. After the phone calls, they've lost confidence in their assessment.

With these results, Markus retires to his office for now. Fortunately for BusinessBooster and SME IT Inc., Markus is looking on the bright side. Although the complete results of the meeting can be summed up as "We don't know," he (and the team) learned something: They have many knowledge gaps. He decides to close these knowledge gaps one by one.

Also, the meeting showed him something else: They can't solve the problem alone before SME IT Inc. runs out of money. So, he makes an entrepreneurial decision. Although money is tight, he decides to seek external support and thus increase the chance of achieving a real improvement in BusinessBooster and the overall situation.

2.3.5 Ergonomics versus Usability versus UX at Healthyfy

The team at Healthyfy, our mediation portal for health experts, has a few advantages: funding for 11 people for 18 months in advance and an additional development budget—things not every startup can boast. The team is excellently positioned and covers the strategic, technical, and commercial sides of the business well. Having a well-established parent company with a large sales team within reach is also a luxury from a start-up perspective. Linda knows all this. But she also knows that all this increases the pressure to show something after 18 months. Otherwise, she will have to answer to why nothing came of these resources despite such excellent starting conditions.

Linda goes into the first substantive team meeting with optimism. After all, she comes directly from a successful start-up and is familiar with modern product development methods. Therefore, Linda uses the first meeting to take stock. She wants to work out with her team what their product needs to do from the user's point of view and where it can inspire. She wants to consider both user groups from the start because she is sure this approach is the only way the product can work.

In the first meeting, it quickly becomes clear that the two user groups are quite different and, thus, have disparate requirements. According to the team, patients would use Healthyfy almost exclusively at home. Mobile devices are the main focus for this user group—above all, smartphones. The use itself, it is assumed, lasts rather briefly—just as long as a typical doctor's visit.

The team sees ergonomic challenges above all when it comes to showing injuries or illnesses. Is it possible for the patient to position the smartphone in a way so that the doctor can see the affected body part sufficiently well? What happens if a user uses a fixed PC instead of a smartphone? Does this result in wild contortions in front of the webcam? Will it still work if people are restricted in their movements, for example, due to age or an injury?

On the physicians' side, the team assumes use at the desk. They imagine the doctor sitting at a table in the doctor's office and having a large monitor, mouse, and keyboard available. Ergonomically challenging, the doctor might also have to perform contortions in front of the screen to see details if the user sends poor-quality images. Since the doctor may use Healthyfy all day, this problem should be avoided.

The *Healthyfy* team has already discussed one aspect of usability in ergonomics. They want to ensure that even the virtual doctor's visit does not take any longer than a physical on-site visit—for both the doctor and the patient. If this time limit were upheld, then at least the patient would benefit from a time advantage, as they would no longer need to travel to and from the doctor's office. But Linda's team also realizes the doctor has no time advantage. Further, for both of user groups, effectiveness could quickly suffer—namely, if incorrect diagnoses are made due to poor images provided by the patient.

At this point, one of the experienced medical device professionals speaks up. He points out to the team that for a medical device to be approved, it must be shown that representative users can use the product under simulated conditions of use without unacceptable risks. Linda realizes that these challenges will still need to be solved.

In the area of UX, the team can think of various points that help the patient. Depending on the illness, the patient could enjoy a positive effect by saving time traveling to and from the doctor's office, perhaps even literally staying in bed. The envisaged function of immediately receiving the prescription digitally and, if necessary, even ordering the appropriate medication directly online could also inspire excitement.

The team also finds potential excitement attributes on the physicians' side. For example, physicians might be excited that care can be provided remotely, which would lower physician and patient exposure to contagious diseases. A digital doctor relying entirely on Healthyfy might even forgo a rented office.

After the meeting, Linda goes over the results again. She notes that some questions were raised. Especially on the physicians' side, the team had concerns. With her experience, Linda can clearly see that she and her team need to learn more about their users, their tasks and requirements, and their contexts of use.

Chapter 3

Usability and UX as Key Factors for Outstanding Products

"There is no existing market for Red Bull, but we'll make it."
—Dietrich Mateschitz (Red Bull co-founder in 1987, before Red Bull became successful)

In our experience, one of the biggest challenges in the corporate environment is convincing management and the (technical) team that usability and user experience (UX) are essential. Unlike in other areas of a company (i.e., in sales) where success can be measured directly, the value of usability and UX can often only be determined indirectly, or a lot of time and effort must be invested to make this value visible.

Usability and UX are sometimes not easy to grasp for many stakeholders within a company, which might involve a change of the well-known way of working after testing and a focus on the user. However, we believe that thinking about how to measure success is the right way to go.

We believe usability and UX design is not an art form that should endure regardless of success but is a means to achieve goals. To assess whether a design measure was successful, you must know the goal beforehand. Only then can you check whether and to what extent success has been achieved. To start, measuring the value of usability and UX design is easier for existing products than for new products since the latter lacks comparable values.

Whether an existing product or a new development, support from the (technical) team and especially from management determines whether usability and UX design will become an issue. Therefore, in this chapter, we'll provide an overview of the current state of measurability and the value of usability and UX design.

On the one hand, we'll present studies that have attempted to quantify the value objectively. On the other hand, we'll show you how different the value can be for different business models. For this task, we'll return to our examples at the end of this chapter. In addition, we'll also chat a bit "from the inside" and report on some customer feedback we've collected in recent years. But all in good time. Let's start with defining a key measure.

3.1 What Is Return on Investment?

Return on investment (ROI) is a term from business administration that describes the profit generated by an entrepreneurial activity in relation to the capital invested. ROI can be expressed in the following formula:

$$ROI = \frac{Profit}{Capital\ invested} \times 100$$

ROI sounds more complicated than it is: Put simply, ROI shows you what you ultimately got for your effort.

The most complicated part of determining ROI is accurately identifying the capital used. In most companies, both money and working time are invested. Accurately capturing these costs and allocating them to individual investments often proves challenging. But even if you can determine the invested capital, the measurement of later profit can be a challenge in the realm usability and UX design. Therefore, in the next section, we'll focus on this part and present various metrics for the business success of usability and UX design measures.

3.2 Metrics for Business Success

How can we measure business success? A wide variety of metrics can be used for this purpose. Which one you choose depends on your perspective and goals. A combination of different metrics is also possible. So again, "It depends."

But since we can't end this chapter right here, let's instead suggest some metrics that can help you measure the success of usability and UX design measures. However, choosing an appropriate metric depends on the nature of your product, system, or service.

This book always looks at metrics that aim to add value through usability and UX design. Thus, the focus is initially on the users or the customers. Of course, you also must consider your company or department, including the goals you have set (i.e., "business success").

A purely customer-related measure could be, for example, to stop charging money for your system, product, or service from now on. This change could have a positive effect on a success metric such as "sales." However, if your company usually makes money from product sales, you clearly have an economic problem at this point.

This economic problem may lead into offering the system, the product, or the service with poor quality or even having to declare insolvency. None of this is in the interest of users or customers—not least because the support is no longer available.

If, on the other hand, your system, product, or service is doing well economically, you can continuously improve the company's performance, which also benefits users and customers.

In summary, even if we focus on the benefits for users and customers, the company's goals must not be ignored.

We can divide measures of business success into three different types:

- *System, product, or service-related metrics* for business success
- *User- and customer-related metrics* for business success
- *Service-related metrics* for business success

We'll present each of these three metrics in detail shortly. However, we want to point out that some of these assignments are ambiguous, and room for interpretation and overlaps between the areas exists.

3.2.1 Metrics Related to Systems, Products, or Services

Metrics related to systems, products, or services say something about the solution itself or its use. In other words, this area measures what the solution can do and how it is used. Examples of system-, product-, or service-related metrics include the following:

- The proportion of functions used
- The duration of pauses between interactions
- The number of completed workflows or purchases
- The number of jumps and terminations within a workflow
- The amount of "consumed" content
- The duration of use or dwell time
- The "path" of the user through the system, product, or service
- The time until the desired action
- The number of desired actions

Imagine, for example, that you manage a website. An appropriate measure could be the "dwell time" of users on this website. Imagine further that, after implementing a usability and UX design measure, you could measure that the average time spent on your website increased from 30 seconds to 2 minutes. But whether that change is a success, or a step backward, depends on your website in the following ways:

- If you run a blog, your goal is to get readers to read the content. A longer dwell time indicates that the content has improved, so more content is now read. A success!
- The interpretation may be different if you operate a website that provides information about public transport departure times instead of a blog. A longer dwell time can indicate that users cannot find their way around the website and must search longer. A step backward!

You notice that the interpretation of data enormously influences the statement. Purely based on dwell time, we could also interpret it as follows:

- Readers find their way around your blog less well due to the usability and UX design measures and take longer to find the content they seek—a step backward!
- Your website for finding public transport departure times has become so clear and fast that users are now comparing other alternatives. In addition, they enthusiastically retrieve additional information that did not exist before—a success!

With this example, we want to emphasize how you must decide which metrics are relevant for you, look at them specifically, and, if necessary, combine them with metrics from other areas to make the desired statements.

The great advantage of system-, product-, or service-related metrics is that they can usually be recorded directly and technically automated. For example, a user's time on a website can be determined automatically using tools such as Google Analytics. At the same time, you also gather a lot of other information, for example, the exact path users take on your website, the number of website visits in general, and bounce rates.

Your users and customers can provide a critically valuable perspective for interpreting system-, product-, or service-related metrics. We'll look at user- and customer-related metrics in the next section.

3.2.2 User- and Customer-Related Measures

User- and customer-related metrics are directly related to the intended target group and their evaluation of the system, product, or service. Therefore, this group of metrics is about testing whether a usability and UX design measure has changed a user or customer rating or their behavior in terms of recommendations.

Typical user and customer-related metrics could include the following:

- The *Net Promoter Score (NPS)*, a metric that shows whether users/customers would recommend a system, product, or service to others
- The number of actual referrals (e.g., via a customer-recommends-customer program)
- The proportion of positive mentions in social networks
- The number of customers acquired
- The number of positive ratings (e.g., five stars in the App Store)
- The cost per customer acquired
- The purchase volume per customer
- The customer potential (e.g., via the customer lifetime value)

User- and customer-related metrics have the advantage of containing a rating by definition. If you increase the number of referrals per customer over time, that's good. Few people would consider getting fewer referrals better.

The same applies with the other user- and customer-related metrics. Lower cost per customer acquired is always better, higher purchase volume per customer is also better, and a higher number of positive mentions on social media is also always better.

So, if you're working with user- and customer-related metrics, you can more directly determine whether the changes are good or bad. Unfortunately, these metrics also have some drawbacks.

One significant drawback—and one of the biggest impediments to usability and UX design being comprehensively established much earlier—is that collecting user- and customer-related metrics is often indirect. By "indirect," we mean that often you cannot measure the actual value yourself but instead must rely on the cooperation of your users and customers.

In contrast to system-, product-, or service-related metrics, which can often be collected automatically, customer satisfaction is measured via questionnaires. Of course, this approach only works if your users also complete the survey.

The same rule applies with some of the other metrics. For example, you can also ask for referrals. But if you don't have an affiliate program or something similar, you cannot objectively measure the number of referrals.

Another challenge of user- and customer-related metrics can be the delayed measurability of a change. For example, if usability and UX design measures increase customer lifetime value (i.e., the total revenue a customer makes with your company for their customer relationship), this change in value is sometimes only found much later.

Fortunately, this limitation does not apply to all user- and customer-related metrics. For example, the purchase behavior per customer or the number of positive mentions in social media can be measured directly and in real time.

Similarly quick and directly measurable are many of the service-related metrics for business success that we present in the next section.

3.2.3 Service-Related Metrics

The final category in our list is service-related metrics. By service-related metrics, we mean the metrics that relate to using *companion offerings* of your system, product, or service. These metrics could include, for example:

- The number of calls to the support hotline
- The duration of hotline calls
- The number of feature requests (requests for certain new functions)
- The number of support tickets
- The number of complaints
- The cost per complaint

- The duration of the complaint processing
- The number of cancellations or reversals
- The percentage of customers who can be helped through self-service (e.g., via a chatbot, frequently asked questions, community help forums)
- The percentage of customers per support channel (e.g., by phone, by contact form, by chat)

With service-related metrics, you can measure everything that happens in the background to provide your system, product, or service the way you currently want to and ensure meaningful contact with users and customers in case of queries.

One advantage of service-related metrics is that they can often be recorded in a technically automated manner. In addition, the interpretation is usually straightforward. For example, few would argue over whether the number of complaints should be increased or decreased.

However, when measuring service-related metrics, these metrics can quickly become distorted when considered in isolation. For example, removing the contact phone number on the website would of course decrease support hotline requests. However, this decrease would not be due to improved usability and UX but simply to removing access.

A clear disadvantage of service-related metrics is that they do not measure the success of the system, product, or service itself but rather the success of the accompanying offerings. However, those offerings do not provide the actual value for users and customers. Thus, the metrics only have a supplementary character. But don't assume they are negligible! For example, handling a complaint often determines the overall rating of a provider. Suppose the customer feels well treated even in the event of a complaint. In that case, this incident can positively affect potential follow-up purchases or recommendations (user- and customer-related metrics).

3.2.4 Choosing the Appropriate Metrics

In only a few cases, a single metric will be sufficient to define the success of usability and UX design measures. However, we would like to guide you when deciding whether a metric is useful in your particular case. Our tip: Take a look at each metric and ask yourself the following questions:

1. What would change for our users or customers if we improved the metric?
2. What would change for our business if we managed to improve the metric?
3. What would change for society if we managed to improve the metric?
4. Especially for all user- and customer-related metrics—but also in general for all "interpretable" metrics (metrics whose change cannot be clearly interpreted as good or bad)—an additional question is: Can we prove with data (e.g., user interviews) that we interpret the metric correctly?

Based on your answers to these questions, you can then select the metrics that have the most significant benefit for your users and customers, as well as for the company and even for society.

To make these guiding questions even easier to understand, let's consider our earlier example of a blog website. Imagine again that you're running the blog. You have identified the *dwell time* metric and set the goal of increasing it.

What would change for your users or customers if we improved the metric? Changing that metric would mean the users spend more time on the website. If the future content were identical to the current content, users would have to invest more time to get the same information. From the user's point of view, this added time does not make sense.

We have already seen that a metric or a chosen goal may not fit well. Something similar would have come out with the fourth question because we would have gotten feedback from user surveys that simply spending more time on a website is not desirable.

In practice, we would now ask ourselves questions two and three: What would change for our business and society if we managed to improve the metrics? If now the answer is "nothing at all," you should seriously consider whether redesigning your website is worth the time.

But if you have good reasons (e.g., you could sell advertising space on your blog more expensively because you can prove that users dwell on the website more), you can dig more deeply and ask yourself what would have to happen to make users want to stay on your blog longer.

This question would then lead to other metrics that you should also consider, in this case, perhaps the "perceived quality of the content on your website." If this metric increases, then the willingness of users and customers to spend more time on your website is likely to increase, allowing you to sell advertising space more expensively. Additionally, you could directly measure the NPS to ensure high satisfaction. So again, a combination of metrics would be helpful to measure the success of the measures.

We hope this point is clear, and you use the questions to identify metrics or goals to optimize your system, product, or service. The keyword is *identify*: When is the best time to start looking at metrics in the development process? We'll address precisely this question next.

3.3 Planning instead of Discovering ROI

Since measuring ROI is particularly challenging regarding usability and UX, we regularly see measures are chosen first without planning ROI. Then, you have a look at what happened or changed. Using the following example of a website redesign, we'll describe the procedure we often observe in various companies.

Our example company finds that customers keep complaining that the website is outdated. As a result, the company decides to set up a new, more modern website. In the following process, an agency screening step occurs, where several agencies are invited to present their ideas for the new website. One of these agencies is subsequently selected because the management likes the proposed redesign.

Then, the entire design process starts with a stakeholder workshop and proposals for the new website, which are then presented to the stakeholders. The design is tinkered with and tweaked intensively, and the agency determines that a modern external appearance, in general, is needed in addition to the great new website.

The original website project is now thus larger and now includes redesigning the logo, document templates, and the general external appearance. Little by little, everything is approved and implemented. The new website goes live, and they are proud of the work.

Over the next few weeks and months, customer feedback comes in, and the new look goes down great. Unfortunately, however, criticism also comes in: Customers can no longer find their way around the website, and although the appearance is consistently praised, the overall impression is negative. The initial internal enthusiasm quickly turns into disillusionment. One or two years later, a decision is made to redesign the website again, but this time properly.

This example may seem slightly exaggerated and far-fetched, but we can assure you that we often find this situation when we are approached. Of course, these requests are usually not about websites but systems, products, or services in general. We are then the ones who are supposed to “get it right now.” But what does “right” mean in this case?

We also start with a stakeholder workshop. (An overview of our approach is provided in Chapter 5, and then details are provided in Chapter 8 through Chapter 13.) Of course, the workshop is also about the visual design of the later website to some extent. But the focus is somewhere else: We want to know what the users or customers want to achieve in the first place. What needs to change, and what needs to stay the same? What needs to have happened at the end of the project for every stakeholder (including the users) to say: “That was fantastic!”?

After each answer, we ask, “And if all that has been mentioned so far was achieved, would you be thrilled, or is there something you’re missing?” until we hear exclusively a “Yes, I would be thrilled, and nothing would be missing!” from all stakeholders.

Based on this approach, we must determine how to permanently monitor the achievement of these goals and how to measure success.

Only then can the concrete work begin. Every design activity is geared towards getting closer to the goals, and the individual design artifacts are tested with users during development. Each user test verifies that we are getting closer to the desired results. In short, “doing it right,” in this case, means identifying the goals at the beginning of a usability and UX design activity, then working to achieve them and measuring progress along the way during development.

Although the approach we just outlined can seem lengthy, it typically is not. We can often identify and set the goals and the most critical metrics in one day. How much time should be spent on goal setting and then measuring and optimizing depends on your goals and aspirations and your system, product, or service.

In general, we use the following rule of thumb: The further apart the goals are, and the higher the *risk* associated with usability and UX design, the more effort should go into planning, measurement, and optimization (see also Chapter 13).

The *risk* we just mentioned can be manifold. On the one hand, you might face an economic risk—for example, wasting resources with an unsuccessful usability and UX design activity or scaring away customers. On the other hand, you might face a physical risk from your system, product, or service that causes people to be hurt if the design is poor. Strategic risk is also conceivable if you develop the product in the wrong direction with your usability and UX design measures and miss the market connection.

One of our projects, for example, was redesigning a user interface (UI) for an emergency landing system in large passenger aircraft. Of course, the project's objective was to increase the operating speed of the pilots so that they could, among other things, decide more quickly about a suitable airfield for an emergency landing.

However, this speed could not be achieved at the expense of decision quality under any circumstances. Accordingly, metrics were defined for both goals and monitored closely in each user test, for example. Both goals were achieved, and this system is now used worldwide in all aircraft of a major German airline. We could not have achieved these goals if we had not determined them in advance and monitored them continuously.

So, defining appropriate metrics to measure success before starting design activities can be immensely helpful and should be done with stakeholders, users, and customers.

3.4 Metrics for Business Success: B2C versus B2B Sectors

In our experience, the selection and consideration of metrics is different depending on whether your solution is in the business-to-customer (B2C) or the business-to-business (B2B) sector. This difference comes mainly from the fact that the influence of usability and UX design on the purchase decision is much more direct in the B2C area than in the B2B area.

Reminder: Customers versus Users

A *customer* is the person who buys a system, product, or service. The *user*, in turn, is the person who uses the solution afterward. These people can be the same person—but they do not have to be.

In B2C, often the buyer and the user are the same person (think of buying a new smartphone, for example). In B2B, in contrast, one person is responsible for the purchase, and others use the system, product, or service afterward. Think, for example, of a smartphone for work that is provided to you preconfigured from your employer.

Let's return to the purchase decision and our statement from the beginning of this section: "Usability and UX design have a more direct influence on the purchase decision in the B2C sector than in the B2B sector." Why is that?

Imagine you're developing a coupon app that offers discount coupons for purchases. The people who download the app from the app store (customers) and those who subsequently use the app (users) are usually the same, as is typical in B2C. The app's usability and design are the direct focus in this case. If neither is met, the purchase is not even made, or the transaction is reversed immediately afterward. If you now improve the usability and UX of the app, more purchases and fewer reversals will result, and you will have directly increased your business success.

This scenario contrasts, for example, with a process control tool with which a company controls its production. In this case, procurement is usually handled by purchasing or—depending on the company size—by the managing director. However, the tool is then operated by the users in production, as is typical for the B2B sector, where the buyer and the user are not identical.

Consequently, both purchasing and users have requirements for this process control tool. These requirements can naturally contradict each other (e.g., "as inexpensive as possible" and "as adapted as possible to the company's production").

The main difficulty, however, is that the different levels of knowledge mean the other side can't easily assess the requirements. Purchasing, for example, might be unfamiliar with the production process itself and, therefore, cannot evaluate the quality of the solution in terms of operation. Even if usability and UX are improved, nothing at all might change from the buyer's point of view. Thus, the sales figures and business success remain the same due to the indirect influence, and the ROI is poor (i.e., the effort was expended, but this effort did not have the desired effect).

This is not to say that you shouldn't do usability and UX design for a B2B product. You should. However, especially in the B2B area, you should follow our advice from Section 3.3. Involve all stakeholders and define metrics that measure goal achievement for all stakeholders.

In a B2B context, you often must define metrics relevant to customers and other metrics relevant to users. Furthermore, when planning metrics, especially in a B2B context, you should consider how to add value for all stakeholders, including customers and users. If you improve usability and visual design with your usability and UX design measure, then that measure pays into the "account" of the users. If, simultaneously, you can prove through user tests that processing with your solution has become 20%

faster and that 30% fewer errors occur, then this change can be converted into money and thus into a measurable benefit for the customer.

You get the idea: The metrics you choose for business success directly relate to your goals. This choice can be strategic.

We hope we have demonstrated the benefits of planning ROI ahead of time, and we hope this demonstration helps motivate you to take on this sometimes arduous journey.

But what can you expect when using usability and UX design to improve your systems, products, and services? Are the increases, as mentioned earlier, of 20% to 30% realistic? We want to address this question in the next section, in which we show what has proven feasible in research and in our projects.

3.5 What ROI Is Achievable through Usability and UX Design?

Choosing metrics and making the ROI visible is not a trivial task. That much is probably evident at this point. This also explains why, although there is broad agreement that usability and UX design can lead to improvements, only a few studies attempt to quantify the value of usability and UX design objectively. In the following sections, we'll mainly refer to an overview study and our empirical values to give you a first impression.

3.5.1 Studies on the ROI of Usability and UX Design

One of the best-known and most extensive studies on the topic of ROI of usability and UX design is published by Jakob Nielsen and his team from the Nielsen Norman Group. In the now fourth edition of *Return on Investment for Usability*, the team examines the ROI of usability measures and looks at their costs and benefits (Nielsen et al.).

On the cost side, the Nielsen Norman Group concluded that, on average, about 10% of the entire budget is used for usability measures in a typical project. However, up to 20% is spent, depending on complexity and risk ("How serious is the influence of poor usability on the result?"). The cost side is thus clearly outlined. But what is the benefit of spending 10% to 20% of the development budget on usability measures?

This study examined 72 projects (mainly websites or website-like products) to answer this question. In each project, information was available on the original design and the redesign, including usability measures, with the usability measures accounting for about 10% of the total project budget.

A total of four metrics were compared to evaluate the before and after objectively:

- The extent of use of specific (sought-after) features.
- The productivity of the users on the website.

- The number of website visitors.
- The number of sales or the conversion rate. (A *conversion rate* is the percentage of website visitors who perform a desired action. So, if 100 people visit a website and you want them to subscribe to your newsletter, your conversion rate would be 8% if 8 of the visitors subscribe.)

So how have these four metrics changed? Let's start with the less interesting metrics: Usage of the coveted features improved by a whopping 174% on average across the 72 projects. *One hundred and seventy-four percent!* That's quite a lot. But okay, feature usage may be significant for users, but what does this mean for you and your business?

Let's look at the other metrics: User productivity has improved by 112% on average. Also, very impressive. Now, of course, you might ask whether this improvement occurred because the website was only of interest to a smaller but more specialized clientele after the measure. The answer to this question is quite clear: no. During the same period, the average number of visitors increased by 91%. This metric means that almost twice as many users now visit the website than before the redesign.

Now for the crucial question: Is it worth it for you and your company? You can assume the answer is yes: On average, the redesign led to an increase in sales or a conversion rate of 87%. Together with almost doubled the number of website visitors, these metrics are outstandingly good values.

Let's now consider a calculation example: If we assume that a website currently has 100 visitors per day and a conversion rate of 5%, then in absolute figures, we would have five conversions per day (5% of 100 visitors). The 87% improvement in the conversion rate causes the conversion rate to improve from 5% to 9%, rounded down (5% × 1.87 = 9.35%). Instead of the 5 conversions, we would already have 9 (9% of 100 visitors) due to this improvement. In addition, however, we also increased the number of visitors by 91%. Instead of the original 100 visitors, we now have 191 on the site daily. Combined with the improved conversion rate, this metric adds up to 18 conversions (9.35% of 191 visitors equals 17.85 conversions). With 10% of the project budget invested in usability measures, the original 5 conversions in our example could be increased by approximately 360%. This feels like a very well-invested 10%.

Although these figures are clear and outstanding, the study's authors estimate that around 80% of all redesign projects are managed completely without usability measures or a usability budget. In addition, the authors warn that the best possible results may not be achieved with a usability budget of 10%.

The authors estimate that the best possible results are achieved when 50% of the total project budget is invested exclusively in usability measures. This cost includes user research and user tests—but not the related design activities. All design activities and further development work are included in the remaining 50%.

If you follow this line of reasoning, the improvements of several hundred percent we've shown are only the tip of the iceberg.

3.5.2 Feedback from Our Customers on the ROI of Usability and UX Design

From our own practice, we can't serve you with robust studies at this point, but we can share some of our empirical values and customer feedback. The fact that most customers are not allowed to report to us on internal metrics in detail does not make the evaluation any easier.

But we actively solicit feedback to understand what working with us has brought. We focus more on measurable results and less on purely subjective opinions—although those opinions are valuable too. Sweeping feedback like "Since we started working with you, our NPS has gone up, and the number of support requests has gone down" are quite pleasing to us. Unfortunately, they don't help us answer the question of what exactly the ROI for our work is.

However, some of our clients closely measure what happens to their solutions during and after joint projects. As a result, we've been privileged to experience the following result: The outcome of our projects is often the translation of the UI design into a style guide and a click dummy based on user research and design iterations. (An overview of our approach is provided in Chapter 5, and then details are provided in Chapter 8 through Chapter 13.)

A *style guide* is a document that tells our customers which UI elements exist, how they look, and how they are arranged (see also Chapter 11, Section 11.12.8). A *click dummy,* in turn, is a prototype of the later UI that shows the workflows and, if applicable, animations (see Chapter 11, Section 11.9).

Therefore, at the end of the project, we hand over to our customers an already tested and user-validated "construction manual," which is then used to implement the UI efficiently. As we already know, efficiency in implementation measures the success of the usability and UX design measure. The investment is worthwhile if we significantly shorten the implementation time and thus reduce development costs. In addition, this approach creates the certainty that the product and the UI will work well with real users, thus avoiding expensive corrections with the development team.

From feedback from various customers, we have learned that this approach speeds up the entire product development process by a factor of 5 to 10, depending on how much influence the user research has on the previously planned functional scope. When user research is done for the first time, it regularly turns out that the users do not need some features classified as necessary by the manufacturer, which also reduces development time significantly.

Using usability and UX design can also significantly increase the productivity of a system. A customer of ours, a manufacturer of security systems at airports, provided concrete figures on this productivity increase. After the redesign, the number of passengers screened tripled compared to the existing system. In addition, the simplified interfaces reduced training time before personnel could be deployed to the stations by around 25%. Comparing the quality of the checks between the old and new systems showed no change.

Measuring success is more effortless for B2C apps that can be downloaded from an app store. In this case, after a single iteration (testing the existing variant and a subsequent redesign focusing on usability and UX), we could observe average improvements in the app store rating of one star. The improvement is less for apps with a high rating; for apps that start poorly, more positive improvement is possible.

With that, we close our toolbox again and jump to the examples to explore the topic of metrics in more depth.

3.6 Case Study: Measuring ROI at moveHUB

Let’s look at our first example—moveHUB, our B2C mobility app for every occasion. At this early stage, deriving the desired metrics is still extremely difficult. Since neither the solution nor the final business model is in place, we can only guess what moveHUB should consider a business success.

A typical goal at such an early stage would be to attract new users, for example. Since we only have an idea so far, acquisition would first be limited to people submitting an expression of interest and saying, “Yes, I can imagine that I want to use that.”

The logic behind this acquisition: People’s interest creates a user base. With an existing user base, various business models can then be realized. However, if no users exist or far too few, we have a fundamental problem.

Fortunately, Matthew recognized this problem—unlike many startups—and approached Emma and Anthony. Instead of continuing with the app’s programming, the three want to find proof that their idea has potential. This step makes sense because only with this knowledge is further development worthwhile. If they could see the evidence, finding investors would also be easier—after all, the business model would no longer be based on gut feeling, but instead on data.

Later, an interesting activity for moveHUB could be to see if users are actually gained to verify the same assumption. So, if an app already exists, the three should look at the number of downloads and actual usage numbers. Possible further metrics would be the rating in the app store and support requests in case of technical difficulties. If a business model is already in place, the three founders should also look directly at revenue and profit as metrics for business success.

However, calculating the ROI for usability and UX design measures for moveHUB would be difficult initially. Since the application is entirely new and no previous version exists, no comparison is possible. Without a “before” version, whether and to what extent an increase has occurred is unclear. Nevertheless, the absolute score of the later application might be appropriate to have a metric for success. Alternatively, the values for customer acquisition or rating in the app store could be compared with the average values of similar applications if data is available.

Further into the future, if the app has been available for some time and is being further developed, they can compare it with the previous version. In this case, the three founders could determine the ROI by comparing the metrics from an older release with the latest release.

3.7 Case Study: Measuring ROI at BusinessBooster

For our B2B application BusinessBooster, calculating ROI is much easier than in the previous section for moveHUB. We already have an existing product, and all data can be compared directly.

Remember that our initially successful software is getting on in years and sales figures are steadily going down. Our most critical customers did not receive the first idea of a technical revision very well. Due to the current situation, the focus of Markus Meier and his team is on revising the usability and the UX as well as on optimizing the visual design. From this effort, he hopes that the sales figures for BusinessBooster will increase significantly and that more people will adopt the trial version in their ongoing operations.

Additionally, Markus would like fewer complaints and negative feedback about the outdated software. In the best-case scenario, the number of support requests related to difficult or impossible information will also be reduced. Should this effort succeed, Markus would be highly excited.

If you ask Markus about his ideal, summarized at a high level, the result would be that more people buy his product, use it longer, and that less support effort for customers is required. To strengthen the company’s position in the market, the team is currently discussing whether a type of partner program could be suitable for obtaining recommendations for the company’s software from existing customers.

Therefore, we now have concrete goals but have not yet quantified targets at this point in the process. The company now has a reasonable basis for finding metrics for business success in the next step. We touched upon some metrics earlier, but let’s present them again clearly.

BusinessBooster might consider the following metrics in this situation:

- The sales figures of BusinessBooster, the overriding goal because the falling sales figures are the primary motivation for the redesign
- The number of downloads of the trial version (if you get more people to try BusinessBooster, the chance of gaining a customer also increases)

- The percentage of companies that used the trial and became a customer or not a customer
- The number of complaints and feedback on the visual design
- The amount of positive feedback about the functions and the visual design
- The number of support requests
- The effectiveness of the affiliate program, which would require, for example, information on the number of referrals that led to a trial download and purchase
- The use of the software itself, that is, via usage statistics about core functions

We could continue with this list for quite a while, but these examples should suffice.

Collecting those metrics before and after usability and UX design measures provides one part of the ROI calculation (*profit*). The other part is the invested effort, which at BusinessBooster is a mix of in-house resources and external service providers (*capital invested*).

3.8 Case Study: Measuring ROI at Healthyfy

A look at Healthyfy's metrics for business success is fascinating. Because this platform addresses both patients and physicians, the *Healthyfy* portal is both a B2B and a B2C product. As a result, metrics from both areas are relevant for checking whether the usability and UX design measures have the desired effect.

Since Healthyfy is primarily intended to be self-service, the number of accounts successfully created relative to registrations started is an exciting metric for both areas. Many cancelled registrations would be a bad sign and a starting point for further improvements.

Moreover, the portal is only attractive if physicians and patients are available in sufficient numbers. This user base is mutually dependent for Healthyfy since patients do not come to the platform if no physicians are registered, and physicians do not use it if there are no patients. To solve such a "two-sided market" problem, artificial incentives are often created for one or both sides to register early. Metrics could include, for example, the absolute number of registrations for both sides.

In addition, the team could measure the encounter between the two target groups (e.g., the number of appointments arranged) as this encounter is where the value is created for both sides.

Linda and her team must also prove to the parent company that the spin-off of Healthyfy is worthwhile. Therefore, several metrics will be selected that allow statements about the willingness to use such a portal and to pay for it (on one side or the other).

In addition, the change in the values over time—above all, the growth in the number of users—will be very much in focus since the benefit of a platform is highest when many people (in this case, doctors offering the service and sick patients) use the service.

In summary, Healthyfy's goals are also the basis for selecting the appropriate metrics, which can be diverse. As with BusinessBooster, the chosen metrics can be compared between older development versions and the current development version to assess progress as soon as it is available. Together with the invested effort, Healthyfy can calculate a ROI.

Chapter 4

How to “Do” Usability and UX Design

“Many roads lead to Rome.”
—Jean de La Fontaine (writer)

In recent decades, many approaches have evolved to help you develop products, systems, and services with good usability and user experience (UX) design. In this chapter, we would like to introduce you to various approaches that can be used to optimize usability and UX. We selected these approaches primarily based on their popularity, which ensures that you can easily participate in discussions with other usability and UX professionals.

In their daily work, our customers use all the approaches we explore in this chapter. In addition, we have developed our approach over the years, combining elements of the other processes that work best for us.

We’ll describe these approaches so that you get a sense of their general procedures and processes. But we won’t go into too much detail to avoid confusion. For each approach, other sources—standards, books, or articles—will help you understand a particular process in more detail. You’ll notice that many approaches have similar roots and often differ only in the details. However, this granularity makes identifying the essential elements of usability and UX design easier.

Derived from the commonalities of the processes, we’ll present our minimum requirements for a meaningful usability and UX design process at the end of this chapter. Building on these minimal requirements, we’ll explain our process further in Chapter 5 and then get into the details from Chapter 8 through Chapter 13.

4.1 The ISO 9241-210 Standard: Human-Centered Design for Interactive Systems

Probably the oldest and most widely used process in the usability and UX world comes from an International Organization for Standardization (ISO) standard. The standard we are talking about is ISO 9241-210, which bears the charming and catchy title “Ergonomics of human-system interaction—Part 210: Human-centered design for interactive systems.”

The standard was first published in 1999. But even today, after more than 20 years, the basic idea of the standard remains the same, and the approach is universally applicable. To be honest, many later approaches build on the basic concepts of this standard. The similarities among the various processes are obvious, so we want to jump into this standard immediately.

If you were to buy the standard, around 40 pages would introduce you to the human-centered design of interactive systems. We won’t the process in such detail in this chapter. Nevertheless, we want to introduce you to the basic idea of the process since it is so ubiquitous—including, by the way, in our data-driven UX design (3DUX) process, which we’ll introduce in more detail in Chapter 5.

The basic idea of this ISO standard is to develop interactive systems (or products or services) that have good usability. To achieve this goal, system users with their needs and requirements are placed in the focus of the development. The goal of the standard is to ensure that with the help of knowledge in ergonomics, human factors, and usability, suitable designs can be created by the reader.

Apart from all the technical terms, it sounds pretty simple. And if we are honest, it is. The charm of human-centered development is that the basic principles are easy to understand. The difficulty is that actually mastering them takes a lot of time. Even with a lot of experience, however, usability and UX design remains a discipline based on exchange and mutual consultation because many solutions are possible, which, by the way, is the situation we love the most.

Human-centered development is a separate procedure you can easily integrate into other processes. Whether the technical development of your system, product, or service is conducted in a waterfall model (linear approach with steps that build on and follow one another) or through agile development, you can integrate a human-centered design process.

The fundamental prerequisite is that the system, product, or service design must be based on a comprehensive understanding of the users, tasks, and working environment. However, theoretical knowledge that can be gained at the desk through research (keyword: *desk research*) is insufficient. You require the direct involvement of users in the design and development of the solution. You must collect feedback, through various methods, from users regarding the design solutions. This feedback is then incorporated into the revision of the solution. Therefore, considering users’ requirements, a product, system, or service is iteratively created, and the overall UX is optimized.

To enable this approach, ISO 9241-210 assumes that a team with interdisciplinary competencies implements the design. The required competencies vary depending on the system, product, or service. Therefore, for your project and company, you must select which competencies are needed in the specific case. From a high-level view, you can incorporate expertise from different areas into your project. The following perspectives could be helpful:

- Strategic view, for example, business analysis, system analysis, and market analysis
- Development perspective, for example, hardware and software development
- Marketing perspective, for example, marketing and sales
- Design perspective, for example, occupational science and ergonomics, usability, accessibility, product design, and user interface (UI) design
- Expert perspective (e.g., users, experts with knowledge in the application domain)

All these different points of view can help you design a successful system, product, or service. At what point and to what extent you include these viewpoints is up to you. But from a purely formal point of view, you can only have a human-centered design process if you regularly and actively bring the expert perspective—that is, the users' perspective—into your project. Therefore, this viewpoint is not always a free choice but, as is often the case, a duty.

The human-centered design process includes a total of six steps, with the first and last steps defining the start and end of the process, respectively; an overview is shown in Figure 4.1.

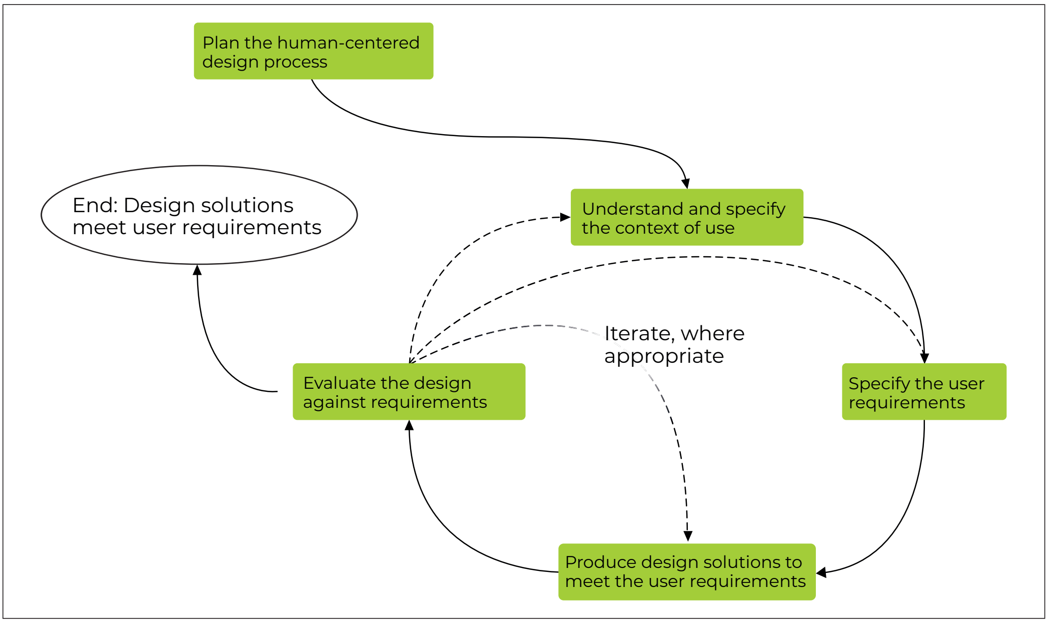

Plan the human-centered design process
End: Design solutions meet user requirements
Understand and specify the context of use
Iterate, where appropriate
Evaluate the design against requirements
Specify the user requirements
Produce design solutions to meet the user requirements

Figure 4.1 Human-centered design process according to ISO 9241-210.

4.1.1 Step 1: Plan the Human-Centered Design Process

The first step is to plan the human-centered design process. You must decide how much effort should be put into achieving good usability and UX. The standard provides the following three core questions to orient you:

1. What is the relationship of usability to the purpose and use of the system, product, or service?
2. What risks can result from low usability?
3. In which (economic) environment does the development take place?

However, no further assistance is given in determining the scope. So here you're on your own with your conclusions.

Once you have determined the scope of human-centered activities, you must plan their content by following these steps:

1. Identifying suitable methods and resources
2. Defining procedures for integrating human-centered development into other development activities
3. Defining responsibilities
4. Defining information and feedback channels
5. Defining milestones
6. Determining the timing of the human-centered design process

With all this information, you can embed the human-centered design process into the overall development process. This step is necessary because usually no finished system, product, or service exists at the end of a human-centered design process, only a "construction manual." These instructions subsequently accelerate implementation considerably: in the case of waterfall planning, the implementation of large parts of the overall product and, in the case of agile development, the implementation of individual building blocks of the product.

4.1.2 Step 2: Understand and Specify the Context of Use

No system, product, or service is used independently of the organizational, technical, social, or physical environment; it is always part of that environment and is influenced by it.

Therefore, the second step is to analyze the current *context of use* to determine the future context of use afterward. (For a definition of "context of use," see Chapter 2, Section 2.2.5.)

A good starting point for understanding the context of use is to look at an existing system or similar systems. Only a few systems, products, or services are entirely new and unique. In most cases, you'll find that users are already solving the same problem or task differently today, which is where you can start. If you already have a solution on the market, you can also start there.

Examining the current situation helps understand how users solve their tasks currently. This examination allows you to discover the strengths and weaknesses of existing solutions and determine what the future context of use should look like. You need this knowledge for the next step. To ensure that the derived knowledge about the context of use is available for the following steps, ISO 9241-210 recommends preparing a *context of use description*. To ensure that this description is functional, the following questions should be answered at a minimum:

- Who are the users and the stakeholders (see the definition in Chapter 2, Section 2.2.4)?
- What are the characteristics of these users and user groups?
- What are the goals and tasks of these users?
- In which environment will the system be used?

How you document all this information is up to you. Our recommendations for implementing and documenting a context of use analysis can be found in Chapter 9.

4.1.3 Step 3: Specify the User Requirements

Based on the context of use analysis, you know where, by whom, and why your system, product, or service should be used. Now, the next step is a matter of formalizing this knowledge and making it testable. To achieve this goal, formulate user requirements in this step. In our human-centric approach, at this point, you might specify requirements that have little to do with the UI itself but are requirements from the organizational context. From our perspective, a core element of human-centered design is that you're not developing an app, software, or other application—you're creating a solution that solves your users' problems and tasks. If the system, product, or service becomes "bigger" due to the insights gained from context of use analysis, then so be it.

When deriving user requirements, you can consider users on the one hand and other stakeholders on the other. You can derive user requirements from the context of use and other sources. These sources could include, for example, legal or regulatory requirements, standards, or organizational processes.

When you collect these user requirements, you'll notice that conflicts may exist: Not all requirements from different sources can be met perfectly. Sensible weighting is crucial to find the right design later. This weighting should be done in the users' interest and include other factors (e.g., economic efficiency).

The list of requirements is not a list that you define once and then remains static. In most cases, you'll find new, expanded, or more precise user requirements during a project that were not initially considered. As shown in Figure 4.1, the process is iterative. Thus, you may add to this list at any time. To get the most benefit from the user requirements, you should make sure that you observe the following rules when formulating them:

- The user requirements should be auditable. Thus, it should be clear to all parties involved whether a requirement is fulfilled or not.
- Stakeholders should support the user requirements.
- The user requirements should be free of contradictions.

Above all, the *absence of contradictions* in user requirements is a topic we want to emphasize, as it turns out to be particularly complicated in practice. The fact that you involve different stakeholders in deriving the user requirements means that you'll encounter situations in which various stakeholders formulate contradicting requirements.

Let's consider an example: In this example company, discussions arise repeatedly in the development team because of a dispute about a function. One part of the team wants to implement this function, and another part thinks this function is not necessary. In such situations, you must decide which viewpoint you believe. An excellent way to resolve such disagreements is to involve the users. In this way, you can decide afterward which team was "right" or whether perhaps even a completely different idea would be more successful.

4.1.4 Step 4: Producing Design Solutions

Now, since you have a first guess of what makes a good solution, you can start producing *design solutions*. When discussing human-centered design, this activity is usually not yet about technical development. Instead, the point is to find design solutions in this phase of human-centered design that subsequently enable an evaluation by users and stakeholders. The evaluation is then carried out in the next step—but prepared in this step to the extent that we learn something in the iteration.

To carry out an evaluation, stakeholders and users need a concrete solution to evaluate. Suitable examples include scenarios, pictures of individual UI parts, simulations, or prototypes that help users understand how they would later interact with the system, product, or service. However, these functionalities do not have to be technically realized. It is enough to build everything "in the background" as simply as possible.

For example, if you need a chatbot, you don't need to program it. Instead, you can put a human in the next room to simulate the chatbot for user feedback.

So, the critical thing with this kind of human-centered design is that you get to a prototype suitable for user testing as quickly, easily, and inexpensively as possible. Why? Because in the next step, you'll test the prototypes with users and, in most cases, adapt them afterward. The faster, easier, and cheaper you can prototype, the faster you can learn and improve your design solution. As a result, your first prototypes are usually quite simple and become more sophisticated and product-like as you become more confident in the design.

However, an essential step is that you also consult with the later implementers already in the development process. You can save yourself—and them—a lot of unnecessary work if you enter into communication at an early stage. This suggestion applies not only to the final design solution but also to the intermediate steps.

We don't want to go into further detail regarding the design at this point as we focus on the processes in this chapter. Therefore, our recommendations for the design and structure of these prototypes are discussed again in Chapter 5, Section 5.2.5, and in Chapter 11.

4.1.5 Step 5: Evaluating the Design

The final substantive step of the human-centered design process is evaluating the design solution. This evaluation should be performed by and with users in most cases. The reason: All so-called "expert evaluations" (i.e., without users) can give you feedback on individual aspects, but usually not on acceptance and not sufficiently on whether the solution fits the users' tasks. However, both elements are crucial for the later success of your system, product, or service. Only in exceptional cases should you deviate from this rule. Suitable exceptional cases could be when you want to identify serious usability problems early on. Expert assessments are usually suitable for this purpose.

When you evaluate design according to ISO 9241-210, you'll focus on the following activities:

- Getting new information about user requirements
- Understanding the strengths and weaknesses of the design solution
- Checking whether user requirements have been met
- Comparing different variants with each other
- Enabling a later comparison with a reference value (benchmarking)

Various methods are available for evaluating the design solution that are not further detailed in the standard. We present suitable methods for evaluation in Chapter 12. Most of these methods presented include an evaluation by users. However, we'll also explore expert methods without user involvement. In addition, some methods perform an evaluation not selectively but over an extended period instead.

The goal of the evaluation, according to ISO 9241-210, is to understand if something needs to be changed and, if so, guide us towards what needs to be changed and in what direction. So, the evaluation is the step where you'll decide if you're done with your design solution or need to go back into the process. The process can be ended if the following conditions are met:

- No new user requirements have emerged.
- Your design solution matches all known user requirements.
- The overall usability and UX are good enough.

4.1.6 Conclusion

The human-centered design process, according to ISO 9241-210, represents something like the original human-centered design model. In one form or another, you’ll always find the elements or process steps mentioned in this section in other processes and consider them repeatedly. Many of the processes we are about to show are slightly different representations or place different emphases but have the same core. So, don’t be surprised if many of the processes we present later in this chapter look familiar.

4.2 Design Thinking

Design thinking has received much attention recently and has successfully fought its way into large, medium-sized, and even small companies. Searching for “design thinking” on the Google search engine throws up more than 1.1 billion hits in less than a second. *1.1 billion!* It's no wonder that various design thinking models and processes exist on the market, with different numbers of steps and stages.

However, the basic idea of a design thinking approach is always the same: solving problems and developing new ideas that are convincing from the user’s point of view. The assumption with design thinking is that the best solutions emerge when people from different disciplines work together to solve a problem—and do so in a creative environment according to a goal-oriented process.

As with usability and UX design, the term *design* stands for the broader meaning of conceiving and developing—and not only for making things pretty. The roots of the approach thus lie in a more general way of thinking about design, compared to the more technical-cognitive process of ISO 9241-210. As a result, however, both approaches are surprisingly similar.

The solutions designed in the process are oriented to the needs and motives of future customers and users and are tested several times. Design thinking often explicitly emphasizes the interplay of three core aspects that successful solutions and innovations bring with them:

- The benefit for humans
- The technological feasibility
- The economic marketability

However, to keep things simple, for this book, we have adopted the 6-step approach of the Hasso Plattner Institute (*https://hpi-academy.de/en/*), as shown in Figure 4.2. This approach focuses on the benefit for humans (i.e., the first point).

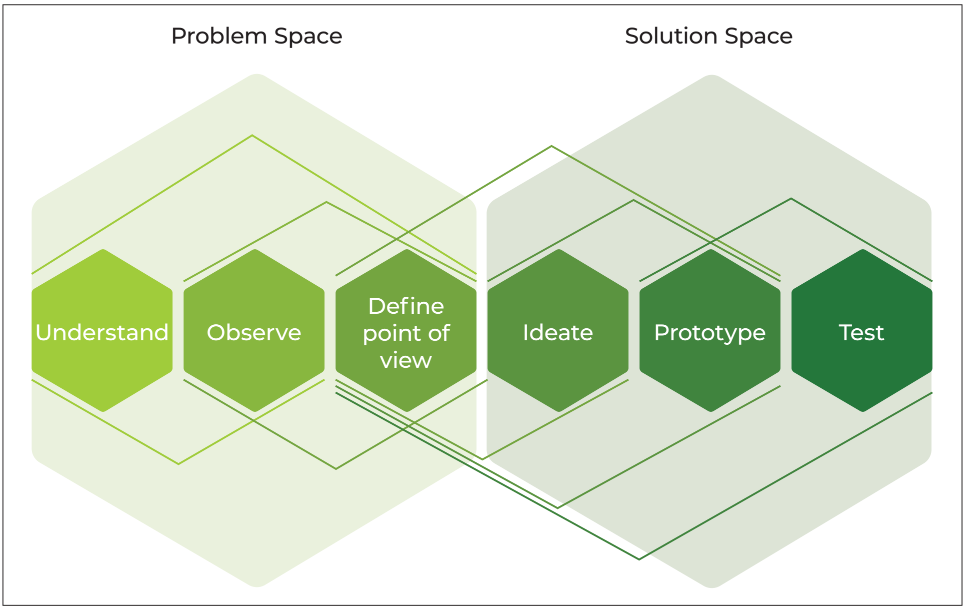


Figure 4.2 Design thinking process based on the Hasso Plattner Institute's 6-step approach.

In this section, we'll present the six steps of the process. However, note that, even if we list the steps separately, *design thinking* is an iterative process run several times that repeatedly jumps between the different steps. In the process, the focus narrows with each iteration, the prototypes developed become more detailed, and the test questions become more targeted.

At the beginning of the design thinking process, the focus is often on the "what." On the contrary, towards the end, the focus is primarily on the "how" of implementation. This division of the design thinking process into two different focal points is also reflected in its steps:

- The first three steps are concerned with analyzing the *problem space*. The aim is to achieve a uniform understanding of the problem for all participants.
- The second half of the design thinking process (steps 4 through 6) is concerned with the *solution space*. Now, suitable solutions are found for the problem from the first half.

This two-part approach is especially beneficial for what are called *wicked problems* (Rittel & Webber, 1973). *Wicked problems* are problems in which many factors contribute to the problem and whose factors can seldom be considered separately from each other. In addition, many different possible solutions are available to resolve these problems.

This approach usually leads to little analysis and immediate solution finding because solutions are more concrete and thus easier to grasp. However, the deliberate separation of the two parts helps conduct the analysis phase thoroughly.

4.2.1 Step 1: Understand

The first step in design thinking is to *understand*. This step is about developing a shared understanding of the problem and the problem space. In most cases, especially when different disciplines and areas are involved, significant differences exist in what should be solved. Therefore, the first step helps clarify the problem and enables the setting of a common focus.

As mentioned earlier and represented by the lines shown in Figure 4.2, design thinking is an iterative process. Thus, once a problem is understood, it is not final and can be adapted repeatedly at various points in the design thinking process. So, you don't make an absolute commitment when you say, "This is the problem we want to solve!" Instead, you decide on an initial direction and a first common understanding. The following quote summarizes this step well. You're welcome to guess who the protagonists are:

> *Character 1: Would you tell me, please, which way I ought to go from here?*
>
> *Character 2: Well, that depends on where you want to get to.*
>
> *Character 1: Oh, it really doesn't matter as long as I…*
>
> *Character 2: Then it really doesn't matter which way you go.*

This excellent quote is from an unlikely source: the Disney movie *Alice in Wonderland*, released in 1951. The protagonists are *Alice* (character 1) and the *Cheshire Cat* (character 2), and this scene has accompanied us in many seminars and lectures for years. The point we're trying to make is this: If you don't have a goal for your design thinking project, that is, if you don't have a problem that you want to solve together, then there is no common direction in which you should go. Then, everything you do is right and wrong at the same time.

You can develop this common understanding by gathering everything you know about the problem at the current time. For this task, search through existing studies, research on the internet, ask internally about existing prior work, and talk to experts in your company. Anything that helps you better understand and narrow down the issue is fine.

You can then put this information together and try to identify the key statements concerning your problem. This task forms the basis for your hypotheses. Hypotheses should be understood as unproven assumptions. So, after the first step, you should have a rough idea of who you want to help with what. You'll determine whether your hypotheses are correct in the next step.

4.2.2 Step 2: Observe

This step aims to gain an even better understanding of the problem space and test your existing hypotheses. If you have worked primarily for yourself and with your team and internal stakeholders in step 1, then in step 2, you dare to step outside and talk to the users and those affected.

The term *observe* covers a whole range of user research tools: from actual observation to questioning to various mixes of methods. At this point, the standard slogan of design thinkers is “Get out of the building!”

In this step, you try to build empathy for the target group and its problems. Through direct contact, you discover whether the problem you want to solve exists for your target group—or whether you just assumed it did. The result of this step can be a confirmation of your assumptions, a slight correction, or a complete change.

Design thinking draws on similar methods to a large extent as the human-centered development process but with a stronger qualitative focus. While classic usability methods are often used for counting and measuring, design thinking is regularly used for observing and describing, especially in the initial phases. Thus, design thinking often exceeds ISO 9241-210 because it enjoys a greater openness and flexibility concerning the problem space and the solution space than is usually the case with pure usability- and UX-related approaches.

The choice of method for user observation or questioning is up to you and the situation. You can find some suitable methods in the Chapter 9 on the context of use analysis. To learn even more methods, we recommend *The Design Thinking Toolbox* by Michael Lewrick, Patrick Link, and Larry Leifer (2020).

After completing this second step, you’ll have obtained data from interviewing and observing users and customers that will help you better understand the problem space and whether your assumptions were accurate.

4.2.3 Step 3: Define Point of View

After you have defined hypotheses about the problem to be solved in step 1 and collected data via observation in step 2, the first moment of truth comes in step 3: You summarize all the collected data and define the point of view. Thus, based on all the previous steps, you can define the *design challenge* and determine your objective.

For this task, first create an overview of all your findings from the observation step. This overview can be created, for example, in the form of *personas* (brief summaries of the essential characteristics of the user group) or *user journey maps* (graphically prepared overviews of the steps taken by the users). Then, check which of your earlier assumptions were correct and which were not.

In practice, a first revision occurs now: You redefine which problem you actually want to solve. You can already think of this step as the first short iteration of step 1—but in the design thinking process, this iteration is listed separately and goes beyond that.

After completing the third step, you have completed the first half of the design thinking process—and thus, the analysis phase is complete. Considering everything you have learned up to this point, you can now define the *design challenge*. The design challenge defines what you want to achieve in the project. Often, the question "How might we?" is asked. This question summarizes the core objective of the design thinking project. An example of the "how might we" question for this book could be:

> *"How might we help people interested in usability and UX design get an overview of the field if we don't have a way to talk to them about it in person?"*

Once you have defined your design challenge, you can start the second half of the design thinking process and look at the solution space, which is precisely where you start in step 4.

4.2.4 Step 4: Ideate

The fourth step, *ideate*, is where the most fun part of design thinking begins for most of us. Now, the focus is on finding creative solutions to the identified problems.

The fact that the fun begins here for many, and that especially tricky problems are often abstract and complex to analyze, regularly leads to this part being started too quickly in practice. The goal is unclear, and thus, the result will not be good. Therefore, at this point, you need the foundation from steps 1 to 3, including user observations, to work in a goal-oriented manner in step 4.

But when you have reached step 4 with a solid foundation, you can get creative and work out solutions with your design thinking team. You can generally use any creative technique you can think of. To learn more about different techniques, look at the literature: For example, the book *Thinkertoys* by Michael Michalko (2006) focuses on creative thinking techniques. Alternatively, the *75 Tools for Creative Thinking* card set by Wimer Hazenberg (2013) can be directly used in team workshops.

We'll introduce you to two central and widely used creativity techniques: brainstorming and brainwriting. Even though both methods are not design thinking specific, we decided to present these methods now to provide you with a starting point for the creative phase. Of course, using both techniques is also possible in all other processes.

Brainstorming

Probably the best-known method is *brainstorming*. In brainstorming, the group works out ideas together and documents them. Elaboration is done by participants simply shouting their ideas into the room, as shown in Figure 4.3. The ideas are then written down by a facilitator and recorded for all to see. The other participants may further develop the ideas. The point is not to have the best idea yourself but to find the best idea as a team. To achieve this goal, creating a space that is as free and creative as possible, in which even wild and unusual ideas can be mentioned without risk, is essential.

Figure 4.3 Brainstorming session at Custom Interactions.

To foster a creative atmosphere, some rules should be followed:

- Try to document the ideas as visually as possible (e.g., a small picture for each idea instead of a lot of text).
- No criticism of the ideas during the collection process—the mood remains positive.
- Don't say, "Yes, but...". Say, "Yes, and..." to show appreciation for and to improve the initial idea.
- Only one person should talk at a time.
- The more diverse the team, the better the results usually are.

Brainstorming is an excellent method to develop new ideas. However, brainstorming can be challenging when it comes to implementation. Especially if your team includes some very loud or very opinionated personalities, or if hierarchical differences play a role, some individuals can dominate the proceedings. We recommend using a moderator who guides the team through the brainstorming session to prevent this from getting out of hand. The moderator ensures that the rules are adhered to—but does not participate in terms of content.

In most cases, such a brainstorming session lasts between 30 and 60 minutes just for the collection. After that, all ideas are out on the table (and not just in our heads) for now. At this point, an essential task is to take your time and not stop right when the ideas start fading. Few problems are so easy to solve that a short brainstorming session with a group of intelligent people is enough—especially when some problems have been around for years. Unfortunately, the ideas that come first are often the ideas that many have come up with before you. So, give yourself and your team time.

If the moderation is not good, the mood of the brainstorming session can tip over. If that's risky for you or if you lack a person who has mastered the art of moderation sufficiently well, then you can switch to other methods at this point without any problems. An excellent alternative to brainstorming is what’s called brainwriting, which we’ll turn to next.

Brainwriting

Brainwriting works in principle like brainstorming, except you set a time for finding ideas, and each participant puts their thoughts on paper individually. In the best case, brainwriting is also performed graphically, as shown in Figure 4.4.

Figure 4.4 Brainwriting session at Custom Interactions.

So, participants have the same time for generating ideas but are not interrupted or criticized by others. Afterward, everyone presents their ideas to the other participants. Usually, a categorization is already carried out so that similar ideas are hung up together to maintain an overview better.

At *Custom Interactions*, we often use two phases in brainwriting: In the first phase, ideas are written down and presented. In the second phase, the ideas are further developed in individual work. The second brainwriting phase is significantly shorter than the first but often brings good ideas to light.

Evaluation of the Ideas Found

Whether you're interested in brainstorming, brainwriting, or any other method, you'll still need to follow up with another phase: the idea evaluation phase. Even though design thinking aims to test ideas quickly and easily, you probably cannot test all of your ideas. In our experience, most brainstorming or brainwriting sessions generate between 50 and 150 ideas, as shown in Figure 4.5.

Ideen
Problem

Figure 4.5 A typical collection of ideas after a brainstorming or brainwriting session.

Of course, not all of these ideas are unique, but they are often at least different enough that a selection must be made.

Many different options exist for selecting the best ideas. For example, you could create an *evaluation matrix* to evaluate all ideas concerning the various requirements, as shown in Figure 4.6. This approach is analytical and gives you a good chance that the most important criteria are considered.

Since many requirements are only discovered during development, this collection is unlikely to be complete at this point. If you want to move more quickly, simple tools such as colored dot stickers can be used. With this technique, each participant has a certain number of votes that may be freely distributed among the individual ideas, as shown in Figure 4.7.

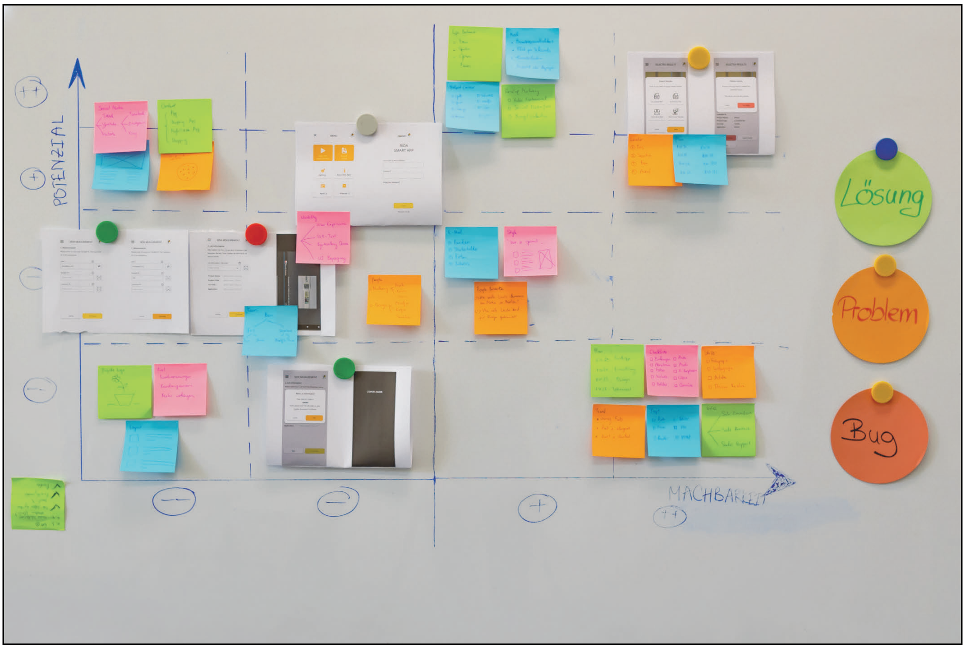

POTENZIAL
++
+
−
--
MACHBARKEIT
Lösung
Problem
Bug

Figure 4.6 Evaluation matrix after a brainstorming or brainwriting session.

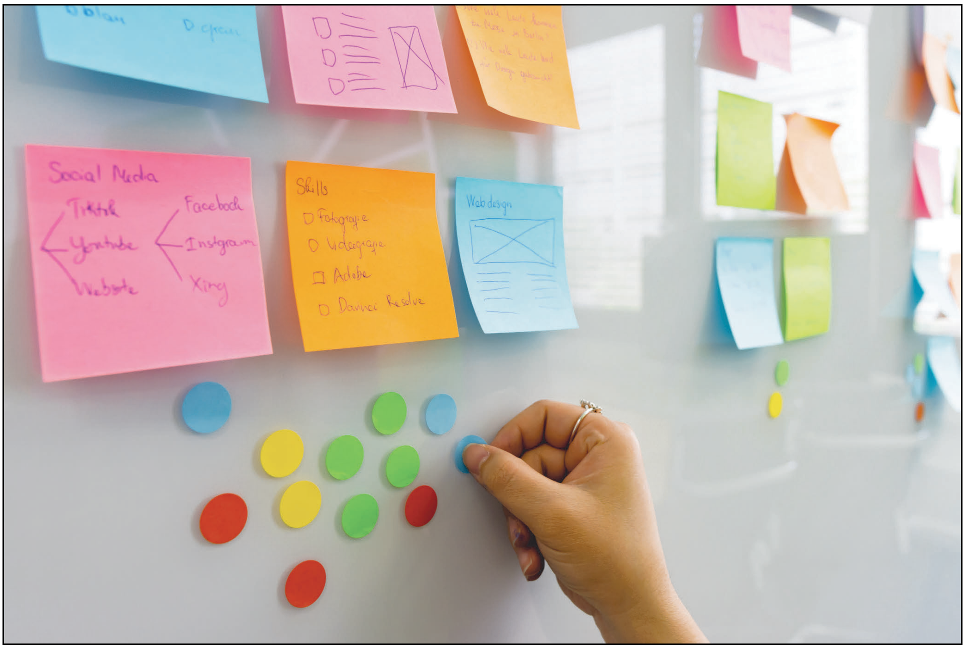

Social Media
TikTok
Youtube
Website
Facebook
Instagram
Xing
Skills
Fotografie
Videografie
Adobe
Davinci Resolve
Webdesign

Figure 4.7 Scoring after a brainstorming or brainwriting session.

At this point, we would like to explore another procedure that allows you to perform a “quick and dirty” evaluation of your ideas.

We found this procedure in a 2018 workbook by Pauline Tonhauser. Her method uses three large, overlapping circles to quickly and effectively evaluate ideas as a team, as shown in Figure 4.8.

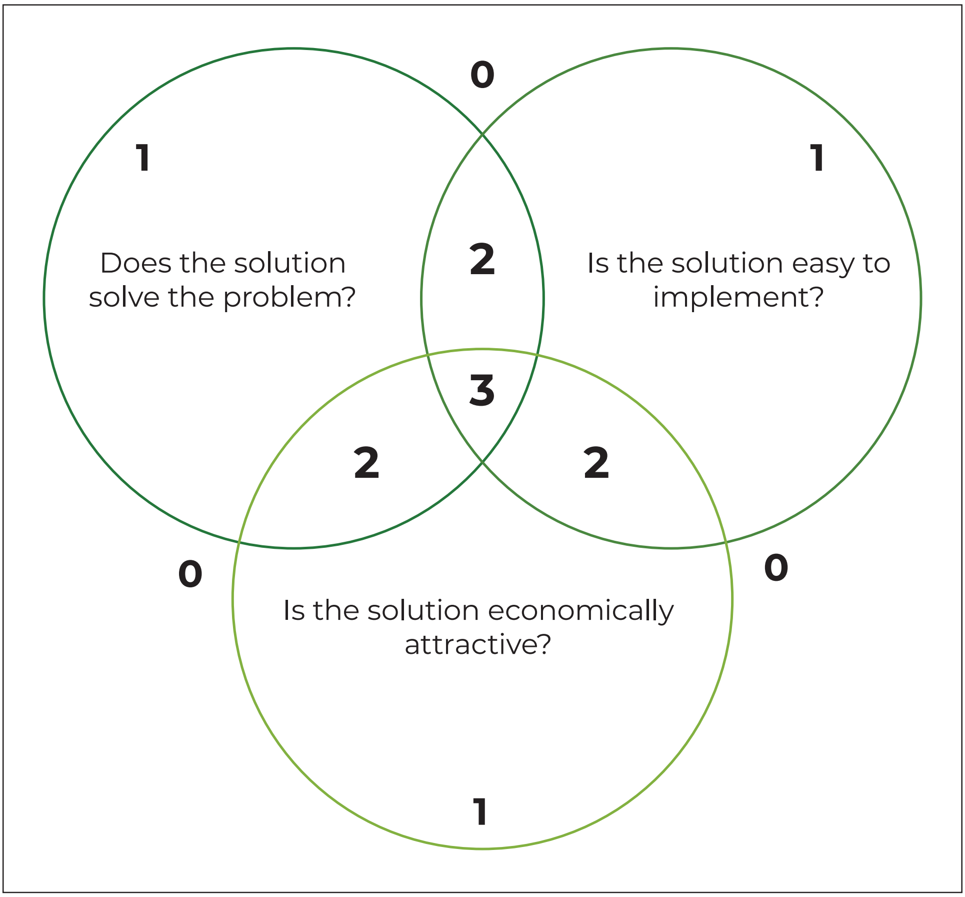

0
1
1
Does the solution solve the problem?
2
Is the solution easy to implement?
3
2
2
0
0
Is the solution economically attractive?
1

Figure 4.8 A simple but effective way to quickly evaluate ideas.

Each circle represents a different criterion. The criteria used for evaluation should be the same criteria you would use later to decide if the solution is successful. Typical criteria that work well include the following examples:

- Does the solution solve the problem?
- Is the solution easy to implement?
- Is the solution economically attractive?

Afterward, each idea (written on a note) is classified. You can physically (or digitally) hold the idea over a circle and ask: *Does the solution* [insert the idea from your piece of paper here] *solve the problem?* If you conclude that the solution solves the problem, the note stays inside the circle. If not, put the paper outside the circle.

Then, still with the same paper, ask yourself: *Is the solution* [insert the idea from your piece of paper] *easy to implement?* Again, you decide based on the question whether the idea remains in the circle or is placed outside.

Finally, take the paper and ask: *Is the solution* [insert the idea from your piece of paper] *economically attractive?* Again, repeat the process.

After this procedure, a note will exist in one of four different areas, which are labeled with the numbers 0 to 3:

- *0:* The solution does not meet any criteria and is utterly uninteresting for further consideration.
- *1:* At least one criterion is successfully implemented by the solution. Nevertheless, this one criterion is too little, and you can ignore such solutions.
- *2:* The solution manages to meet two out of three criteria. You can save these solutions for later.
- *3:* The solution is a bull's eye because it meets all your criteria. Congratulations, you should take a closer look here!

We'll revisit this procedure in Chapter 11, Section 11.4, and also play through some cases where you have no solution or many 3-point solutions. For this section, however, we'll leave it at that and assume you now know your best ideas. You're ready for the fifth step: developing the prototypes.

4.2.5 Step 5: Prototype

In the fifth step of the design thinking process, you develop your best ideas into prototypes. Design thinking is not about technical prototypes that demonstrate technical feasibility but instead about creating prototypes that help the users later experience the product. In this step, a wide variety of materials can be used that are quickly at hand and suitable for visualizing initial ideas. Drawings, modeling clay, modular blocks, paper, and cardboard are typical, but any other material that enables a quick feel for the product is also conceivable, as shown in Figure 4.9.

No matter what kind of prototype is chosen, the level at which the prototype is presented is essential. Since the prototype will be tested with users in the next step, prototypes should illustrate the experience at the level at which the user will later interact with the product.

As a concrete example, if you want to design an app, you would show screens of the app, not the schematic image of the IT infrastructure. As with the human-centered design process according to International Electrotechnical Commission (IEC) 9241-210, the goal is to create a prototype suitable for user testing as quickly, simply, and cost-effectively as possible.

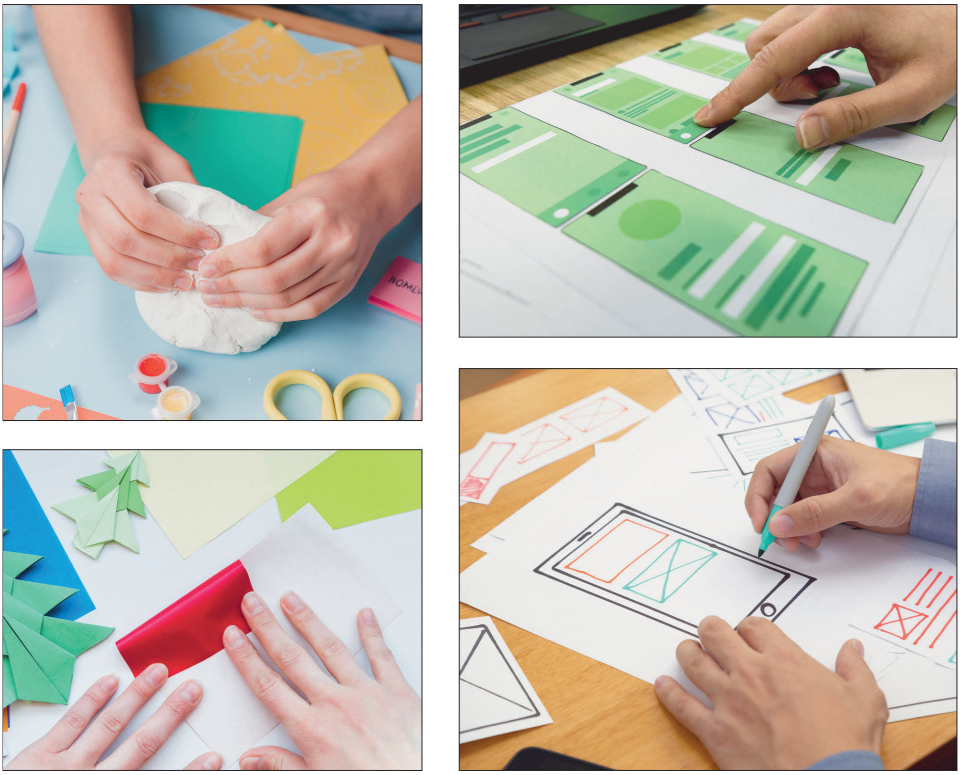


Figure 4.9 Different prototypes from design thinking sessions on different topics.

4.2.6 Step 6: Test

The last phase of design thinking is the testing of the prototypes. In this step, the prototypes are shown to real users. The tests can be rapid and informal or planned in detail and thought through for a long time. No matter how you proceed, the point is to confront people who will use the product with the prototypes and get feedback from them, as shown in Figure 4.10.

In the first iteration, a significant focus of testing and feedback will be on whether the problem has been understood adequately and whether your solution is helpful. In later iterations, the focus will be whether the chosen solutions are well implemented, easy to understand, and self-explanatory.

The end of the test will always be a collection of the problems found. Ideally, you already have some prioritization, so you know which issues you need to address first, and which are less critical. Then, you start the iterative revision process. If you've discovered fundamental problems, jump back relatively far to the beginning. If you've only found more minor issues, you might revise the prototype. Ultimately, you keep doing it until, eventually, your users are happy with the solution.

Figure 4.10 User test in an early product development phase.

4.2.7 Conclusion

In design thinking, the first part is about understanding the problem (steps 1 through 3), and the second part is about solving it (steps 4 through 6). With the help of an interdisciplinarity and holistic approach, design thinking promotes a way of looking at problems beyond quickly jumping to simple solutions. Instead, the design thinking process succeeds in casting the complex process of creative thinking into a simple and systematically comprehensible process that enables a careful consideration of the problem and the solution space. The direct involvement of users ensures that the developed solution does not bypass the market.

4.3 Minimum Viable Product

In his book *The Lean Startup*, Eric Ries (2011) caused quite a stir in the startup scene and contributed to MVP becoming a common term in product development. The idea of the *minimum viable product (MVP)* is that, for startups, the limiting resources are not ideas or creativity but money.

In this context, a startup means not only a (financed) new business but any company or part of a company that is looking for a (new) business idea that works. This search is usually limited in time by the available resources. A startup can easily calculate when the money is out by comparing its account balance with its monthly costs.

However, the time needed to find the business model is always finite and can only be extended if the company generates revenues in the long term. Building up a sustainably functional organization that can fall back on a solid business model for a sufficiently large market is necessary to ensure this longevity.

Therefore, according to Ries, the MVP approach aims to develop a working business model as quickly as possible, with a product at its core that is good enough to create value for customers and users—and make money from it in the long run. So far, so good. But how does an MVP work exactly?

The MVP process aims to develop a product that consists exclusively of the core features needed to be valuable to customers. Everything else is initially left out.

By reducing the scope of functions and, if necessary, ignoring quality, you can bring an MVP to market and make money with it much more quickly than launching a "full-fledged" product that brings many features of higher quality. Thus, if you plan to release a simplified product version first to test the appeal, you'll most likely discuss developing an MVP.

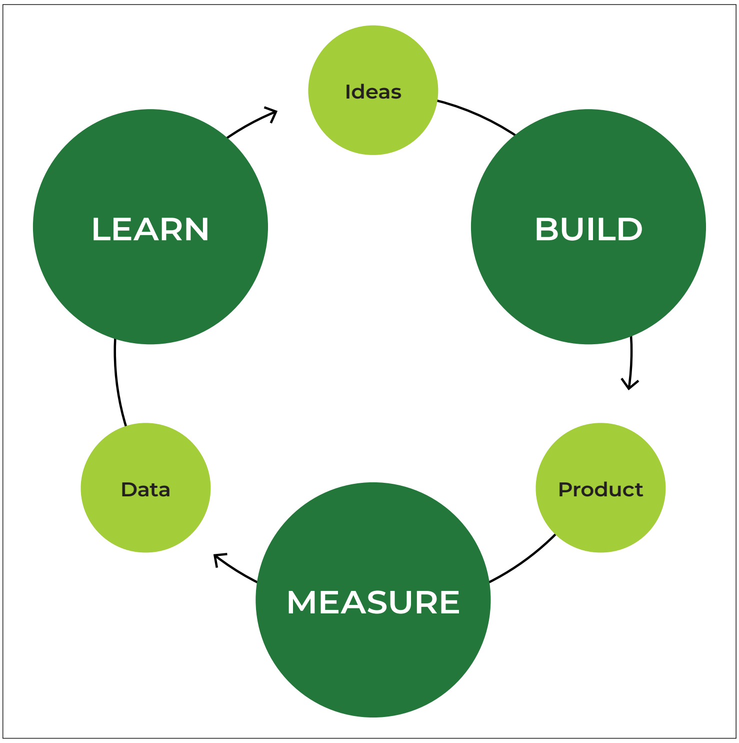


Figure 4.11 The build, measure, and learn steps of the MVP process based on Ries, 2011.

According to Ries, the MVP process consists of three core steps repeated until you have an MVP that customers are willing to pay for. However, unlike the approaches presented, the MVP process does not start with a detailed analysis phase. MVP begins with development, moves on to measurement, and ends with learning from the data obtained, as shown in Figure 4.11. The cycle then starts again. In this section, we’ll closely examine each step, starting with development in the “build” step.

4.3.1 Step 1: Build

The first step, *build*, is about developing something visible and usable by users. This goal does not necessarily mean a finished product. The result could also be a website or a flyer that users can respond to.

Imagine planning an online equestrian supplies store: You guess that a high demand for equestrian supplies exists especially in small towns in rural areas. So now, before creating a complete online store, adding thousands of products, and signing supplier contracts, you should ask yourself how to make your idea easier to implement. The goal is not to set up the first step as “final” but to design it so that the idea is testable.

For example, you could photograph products in an existing equestrian store and put them on a simple website. Then, you could run ads on the website, hand out flyers, or draw attention via banners at equestrian events. The important thing is that you don’t implement anything unnecessary to test the idea. For example, you don’t need a sophisticated store system or actual suppliers.

If an order is received, go to the equestrian store and buy the product retail. Of course, you won’t make a profit. Instead, you’ll make a (small) loss, but you can measure whether your idea is working. In the simplest case, you could even just set up a website where people could express their interest in being informed when the store goes online. You would not realize any sales, but you would have an initial way to gather expressions of interest, which brings us to the next step.

4.3.2 Step 2: Measure

As we have just indicated, the MVP process relies on rigorous reduction in the build step. You build only what customers and users really need. This rule is as true for the first run as it is for each subsequent iteration. On the other hand, the MVP process relies on measuring whether your actions are successful.

Often, success is measured subjectively (“I like it” or “I don’t like it”). If you’re a startup, you need money from customers or investors. Being liked is not a convincing argument. So, what could you measure?

If we think about our earlier chapter on return on investment (ROI), especially Chapter 3, Section 3.2, we explored some possible metrics. Whatever objective metrics you decide on (we recommend something like buyers, registrations, or repeat usage), the metrics must be measurable and measured repeatedly at regular intervals.

In the early days, many now huge companies focused on increasing just one key figure. For Dropbox, that figure was user numbers; for YouTube, video minutes watched; for Airbnb, overnight stays.

But the MVP process doesn't require committing to a single metric. What's important is that you can decide whether the approach you're currently driving contributes to your goal. Only in this way can you successfully initiate the MVP process's third step.

For the equestrian supplies online store, you could, for example, measure the orders received or the order volume per day.

4.3.3 Step 3: Learn

The last step is simply about learning from the data. As in all the other processes we have presented, the MVP process assumes that the first idea will not be the final solution. So, an essential task is to collect data and start experiments in such a way that it becomes possible to learn something from it. In principle, the MVP process is based predominantly on large-scale digital products and therefore checks success mainly quantitatively. But for this to work correctly, you need to have designed your experiments so that you can draw the correct conclusions from them. Let's go through this scenario with our equestrian store example.

Imagine you want to understand whether a general online store for equestrian supplies or a specialized store (e.g., only for riding clothes or riding crops) is better. To test this choice, you could set up two online stores: one that includes many different items and one that provides only riding crops. Then, you could run ads on both sites and see which stores get more clicks and orders. That would be a neat way to go.

But if your time and money are limited, implementing both could be too costly. An alternative could be to set up a single online store that includes both and then create two different flyers. One flyer promotes only the riding crop store, and the other promotes the general online store. Both have unique links; therefore, you can check later which flyer brought more visitors to your site or received more orders.

This approach reduces the effort considerably, but you can still clearly tell which works better. On the other hand, if you always use the same link, then you have the problem that you may have more orders and calls, but you can't say why that is. You could then only conclude that flyers work as an advertising measure.

We hope you take the following lesson with you: To learn meaningfully from the data, you must set up your experiments for one-to-one mapping. You can change one thing at a time or perform targeted comparison tests. Knowing what works well and what doesn't, you move on to the next MVP cycle to create an even better MVP that brings more value to your customers and users.

4.3.4 Conclusion

Unlike the approaches shown so far, an MVP is not a prototype (technically, it could be) but instead is a product that can be purchased or acquired. It may be that users know about the prototype stage. But it may also be that you hide this information well. In any case, however, users should already be actual customers—spending money, providing data, or other information in return for using it.

Furthermore, according to Ries (2011), the point is that you iterate through the three steps multiple times, but most importantly, as quickly as possible. The assumption is that, if each iteration makes you better, you should iterate as quickly as possible. But if you’re a startup with limited time before you run out of money, you need to iterate as quickly as possible to learn as much as possible. In this case, you have the best chance to get an MVP with the available money that's worth scaling afterward. You can achieve this speed by consistently leaving out features you don’t necessarily need in your MVP.

By the way, did the example in the first step also make you think, “Who goes to a real store, takes pictures of products, and puts them online?”—That’s precisely the story of a German founding team that wondered in 2008 whether shoes could be sold online (which wasn’t widespread at the time).

More precisely, the three of them started with photos of flip-flops and set up a simple website. The cell phone number of one of the founders served as customer service. Did testing and improving the business case work out? Why not go to *www.zalando.com* and take a look? Spoiler alert: They made 9 billion dollars in revenue in 2020.

4.4 Double Diamond

The Double Diamond model is another ideal approach to human-centered design projects. This model was developed in 2005 by the *British Design Council* (2007) and has been used worldwide to structure and guide the work of designers.

The British Design Council views the double diamond as a comprehensive framework that describes the design work process and provides core principles and methods necessary to produce good design work in the long term.

The core idea of Double Diamond is the integration of divergent and convergent thinking. In *divergent thinking*, a question is considered in detail, and a topic can be dealt with either very deeply or broadly. In *convergent thinking*, the focus is then placed on individual aspects.

Like design thinking, a wave movement alternates between broad and narrow thinking. The Double Diamond model involves four phases, as shown in Figure 4.12, which we’ll briefly describe.

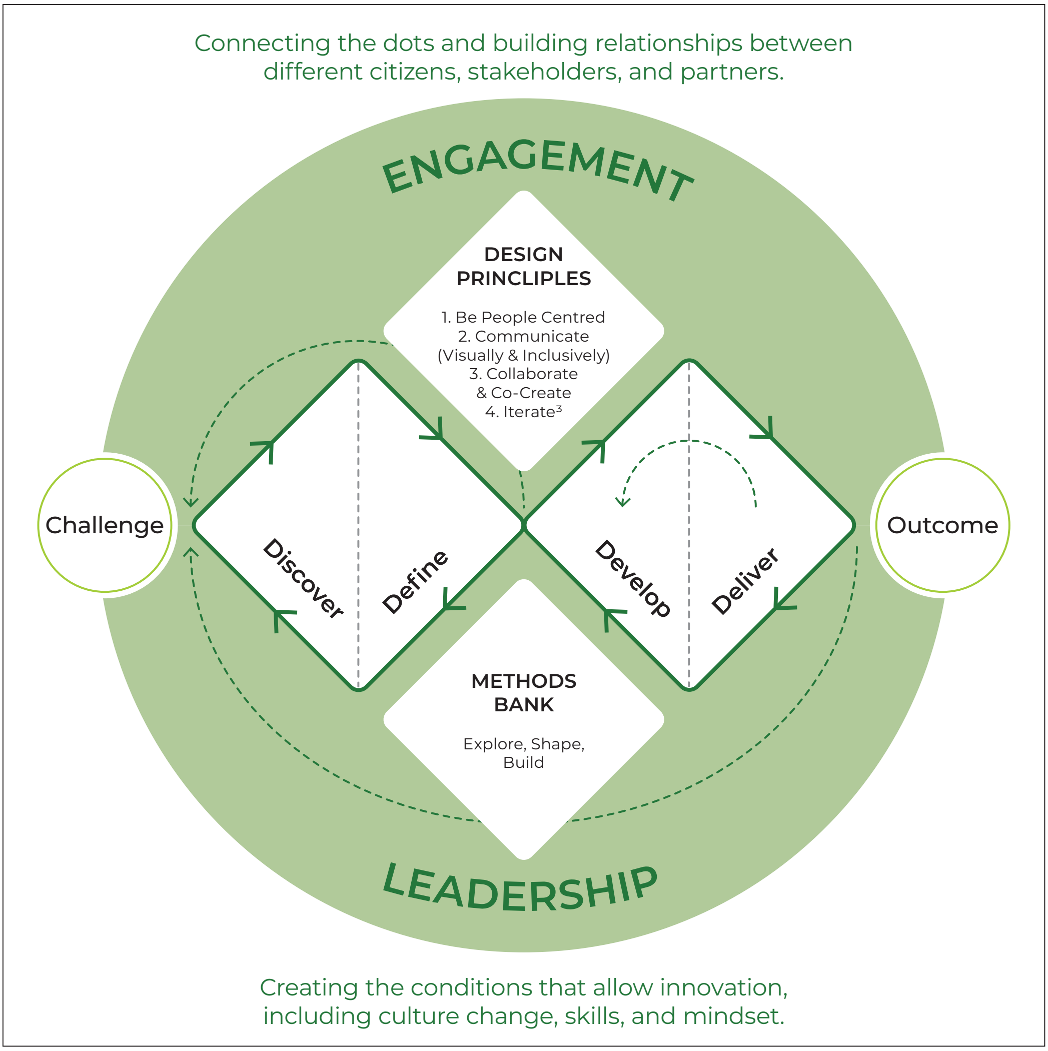

Connecting the dots and building relationships between different citizens, stakeholders, and partners.
ENGAGEMENT
DESIGN PRINCIPLES
1. Be People Centred
2. Communicate (Visually & Inclusively)
3. Collaborate & Co-Create
4. Iterate³
Challenge
Discover
Define
Develop
Deliver
Outcome
METHODS BANK
Explore, Shape, Build
LEADERSHIP
Creating the conditions that allow innovation, including culture change, skills, and mindset.

Figure 4.12 The Double Diamond model. (Source: British Design Council; www.designcouncil.org.uk)

4.4.1 Phase 1: Discover

The first phase, *discover*, is about deeply understanding the problem. The emphasis is on the difference between guessing and knowing. You're not making an educated *guess* exclusively. Instead, you are trying to understand the problem in depth, including its implications for the people involved, and are trying to support that understanding with data.

According to British Design Council, this phase *"involves speaking to and spending time with people who are affected by the issues."* You're not advising; you're researching.

Looking at the model shown in Figure 4.12, you’ll recognize the two diamonds or rhombuses that give the model its name. The discover phase is a phase of divergent thinking because the diamond becomes wider and gets more volume during this phase. So, you can gather a lot of different information and illuminate various aspects.

4.4.2 Phase 2: Define

The second step in the Double Diamond model focuses on *defining* to avoid drowning in the sheer volume of data. You must draw the story you want to believe and follow from the data. Now is when your skill at analyzing data comes in: You must understand what story the data has to tell.

From our experience with many aspiring and experienced UX professionals, we can say that this step is probably one of the hardest. Understanding patterns in data requires a lot of practice and analytical skills. Nevertheless, this step is necessary because, without deriving the pattern, you’ll always remain at a level that makes it challenging to design your system, product, or service.

For example, a typical uncondensed statement would be, “3 out of 10 people I’ve talked to want feature Z.” As long as you don’t understand the underlying need behind the desire for feature Z, you’ll constantly develop what are called *feature solutions*, which can be pretty good—but for which a large amount of uncertainty always remains in the development as to whether they are outstanding.

The *define* step is designed take time to specifically analyze the data and derive the design task from it. The *define* step is a convergent thinking step where you narrow the mass of available information into a few core results. Note that *define* does not mean you can pick and choose what data to use! *Define* means that you recognize the pattern in the data and select the pattern that the data fits. The *define* phase concludes the first diamond of the Double Diamond model.

4.4.3 Phase 3: Develop

After clearly defining and naming the problem in the previous phase, you’re now thinking broadly and looking for solutions. This phase is about working out which possible solutions might be suitable for your problem.

As in many other creative fields, the focus is on collaborative solution finding. “Collaborative” means working with different people, perspectives, and professional backgrounds. Furthermore, inspiration from other places (e.g., similar problems, comparable solutions, parallels in nature) should be sought and integrated.

This creative divergent phase is followed by another convergent phase. We’ll take a closer look at this phase in the next section.

4.4.4 Phase 4: Deliver

In the final phase, *deliver*, the goal is to select and implement the best solution from the many available and possible approaches. The *deliver* phase is iterative. In this step, all solutions are implemented as prototypes. The focus is again less on technical implementation and more on making the solution approach tangible from the user's perspective.

Simple user tests take place to again collect real opinions and data on the solution approaches. All approaches that don't work are removed. All approaches that generally work are further improved, and the best approach is finally (technically) implemented.

4.4.5 Key Principles

In addition to the four phases, the Double Diamond model is based on several underlying principles. For completeness, we'll also briefly mention some of these key principles:

1. Put people first.
2. Communicate visually and in an integrated way.
3. Work together and collaborate.
4. Iterate, iterate, iterate.

Behind the first key principle, "Put people first," is the belief that a deep understanding of the problems, needs, strengths, and hopes of the people involved is the basis for finding successful solutions.

People are visual and can grasp graphical information much faster than textual information. Therefore, the second key principle emphasizes visual communication to create a shared understanding for all parties (both designers and users). This fundamental principle also underlies our own work. We hope you believe us, even as you're reading this book of several hundred pages. Nevertheless, we'll stick to the statement: We don't write down an idea for further development or for a user test but instead we visualize it or build a prototype.

The last key principle, "iterate, iterate, iterate," describes once again that the first solution will not be the final solution and that testing ideas is necessary to detect errors early on to avoid getting stuck on an idea.

4.4.6 Conclusion

The charm of the Double Diamond model lies in its simplicity. True to the core discipline of the British Design Council—namely, design— the model is simple, appealing, and easy to use. The four phases are logically structured and easy to follow but comprise various possible steps and methods. The British Design Council even offers a collection of methods on its website for each phase. More information can be found at *https://www.designcouncil.org.uk/*.

4.5 Usability Engineering for Medical Devices According to the IEC 62366-1 Standard

The field of medical devices is a particular case for implementing products with good usability. Medical devices are special because problems with usability can have much more severe consequences than in the design of most other products.

At the beginning of this chapter, we introduced you to the ISO 9241-210 standard, which explains the human-centered design process. A comparable standard for the area of medical devices exists from IEC standard number 62366-1. This standard is entitled “*Application of Usability Engineering to medical devices*.” The fact that there is a different standard for this area clearly shows that usability engineering plays a significant role in developing medical devices. The medical technology industry is one of the few that require usability engineering.

As a UXer, before you break out in enthusiasm and say: “Perfect, I have to get into that industry!” a little warning in advance: Usability engineering in the medical technology industry is not the same as, for example, usability engineering in the software industry. While IEC 62366-1 takes the definition of usability from ISO 9241-11, it clearly states that it is “only” about ensuring safe operation. The ultimate goal of usability engineering for medical devices is to provide products whose operations are free from unacceptable risks to users, patients, and third parties.

To put it bluntly, operating errors should not cause unacceptable damage. This rule is directly comprehensible for medical devices since incorrect operations can result, for example, in the wrong medication, pain, treatment coming too late, incorrect diagnoses, or, in the worst case, patient death.

A quick side note, as a *usability engineer*, you’re responsible for interacting with a device. Of course, you’re not responsible for the safety of the overall product, but the associated *risk manager* is. Your focus is on the operation of the product and the interaction between people and the product.

As a usability engineer, you’ll encounter an unusual mixture in the medical technology industry: On the one hand, everyone involved knows that usability engineering is done and needed. On the other hand, the primary goal is not a fantastic product with an excellent UX but just getting approval, which really just means checking off requirements from the usability area. A situation, then, in which we often don’t know whether to laugh or cry.

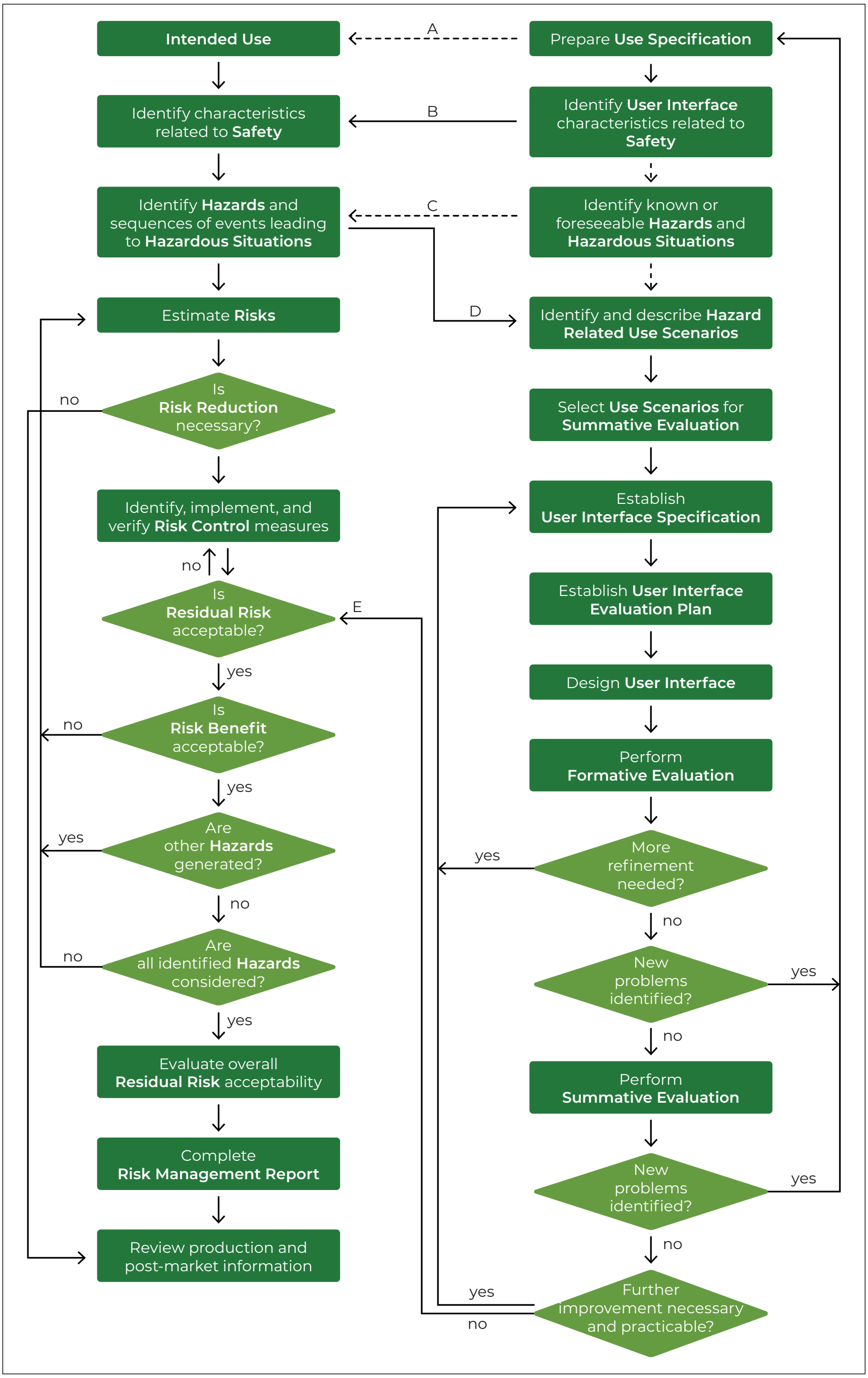

Intended Use
A
Prepare Use Specification
Identify characteristics related to Safety
B
Identify User Interface characteristics related to Safety
Identify Hazards and sequences of events leading to Hazardous Situations
C
Identify known or foreseeable Hazards and Hazardous Situations
D
Estimate Risks
Identify and describe Hazard Related Use Scenarios
no
Is Risk Reduction necessary?
Select Use Scenarios for Summative Evaluation
Identify, implement, and verify Risk Control measures
Establish User Interface Specification
no
Is Residual Risk acceptable?
E
Establish User Interface Evaluation Plan
yes
Design User Interface
no
Is Risk Benefit acceptable?
yes
Perform Formative Evaluation
yes
Are other Hazards generated?
yes
More refinement needed?
no
no
no
Are all identified Hazards considered?
New problems identified?
yes
yes
no
Evaluate overall Residual Risk acceptability
Perform Summative Evaluation
Complete Risk Management Report
New problems identified?
yes
no
Review production and post-market information
yes
Further improvement necessary and practicable?
no

Figure 4.13 The usability engineering process for medical devices according to ISO 62366-1 with the usability part on the right closely linked to risk management on the left.

Of course, plenty of manufacturers strive for an outstanding UX. Various manufacturers combine the required usability engineering with human-centered design in the sense of the processes shown so far. But of course, as in all other industries, an outstanding UX is not required by law.

We’ll provide a brief overview of the process, as shown in Figure 4.13, for a basic understanding of usability engineering for medical devices.

Compared to the other processes, the process for medical devices is much more extensive and has more steps. Many steps involve creating documents and proving that you have done the right things because the medical technology industry is much more regulated and complex than other industries.

Since a medical device can literally be a matter of life and death, as a usability engineer in this field, you must prove your personal competence. Thus, before you start in this field, we recommend that you attend at least basic training and have a document stating that you have participated in those activities. After all, in the medical device industry, “What’s not documented never happened.”

We have decided against presenting the process here in detail, as its focus goes beyond our scope. However, we decided against leaving the process out entirely for two reasons: First, the reference to risk is not so clearly elaborated in any other process, but it is also relevant for other areas, for example, process control. Second, we’ve observed that medical technology is becoming increasingly important, and we would like to contribute to the awareness that different rules apply to it.

4.5.1 Conclusion

The usability engineering process for medical devices involves much more detail than processes from other areas. Many process steps relate to assessing and controlling risks associated with the operation of the medical device. The process is an absolute necessity if your solution is a medical device.

However, if your system, product, or service is not a medical device, this process often will mean you're cracking a nut using a sledgehammer.

The only exception is if there is also a real risk to the user or third parties when using your product. Then it would help if you closely examined how the usability engineering process makes the risk visible and manageable.

4.6 Commonalities among Methods

Throughout this chapter, we have explored some of the most popular processes for achieving design solutions with high usability and UX. Do not consider this selection of processes exhaustive. We’ve only included the models we’ve encountered repeatedly in our daily work and that we think you should know.

Of course, various similar models from other areas aim at iterative improvements to ensure exceptionally high quality. Especially in the quality management area for processes and manufacturing, many comparable approaches can be found that involve a similar mindset, for instance, the *continuous improvement process* (CIP; see ISO 9001 or Witt & Witt, 2006) or *Kaizen* (see Imai, 1986). However, these models are primarily about finished products. Thus, if a model you're familiar with is missing from this book that you feel is essential, contact us at *handbook3DUX@custom-interactions.com*.

Our goal in this chapter was not to prescribe a particular model but instead to make clear which core principles and processes have proven themselves across different areas, times, and approaches. We want to enable you to decide how you want to proceed. To make this possible, we'll highlight the most important common features of the processes we've mentioned. We'll conclude each section with a question to help you decide where your team stands in this area.

4.6.1 Analysis of Customer Needs

In all the processes we presented in this chapter, the analysis of customer needs, as shown in Figure 4.14, occurs at the beginning of development.

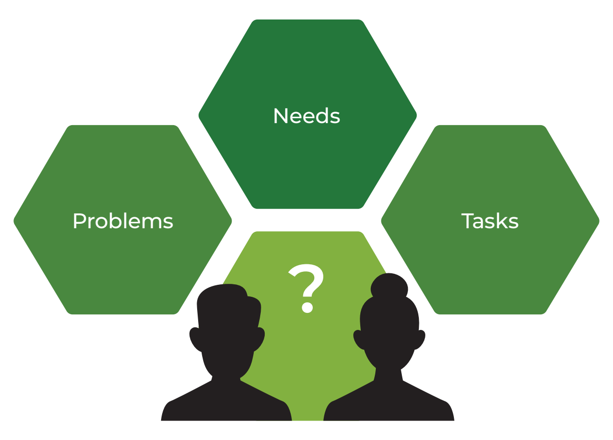


Figure 4.14 Users with (individual) problems, needs, and tasks.

Among customer needs falls a preoccupation with the following questions:

- Who is my customer/user anyway?
- What are the tasks of my customer/user?
- What problems does my customer/user want to solve?
- What challenges does my customer/user face with the solution?

The processes we presented are similar in that a deep understanding of the customer/user and their initial conditions is a prerequisite for the success of a solution. The idea behind this stance is that a system, product, or service is used if a real need exists for the customer/user. Fully understanding this need is the basis for finding a successful solution: *Do you know what problem your product, system, or service solves?*

4.6.2 Iterative Optimization of the Solution

Based on the understanding of the problem, solutions are developed that precisely address this problem. The core question behind every solution is not a technical one like “What technical basis do we use?” “Which GUI framework is used?” or “What does the database look like?” Instead, in the first step, a content-related question should be asked: *Does this implementation contribute to solving the customer/user’s problem?*

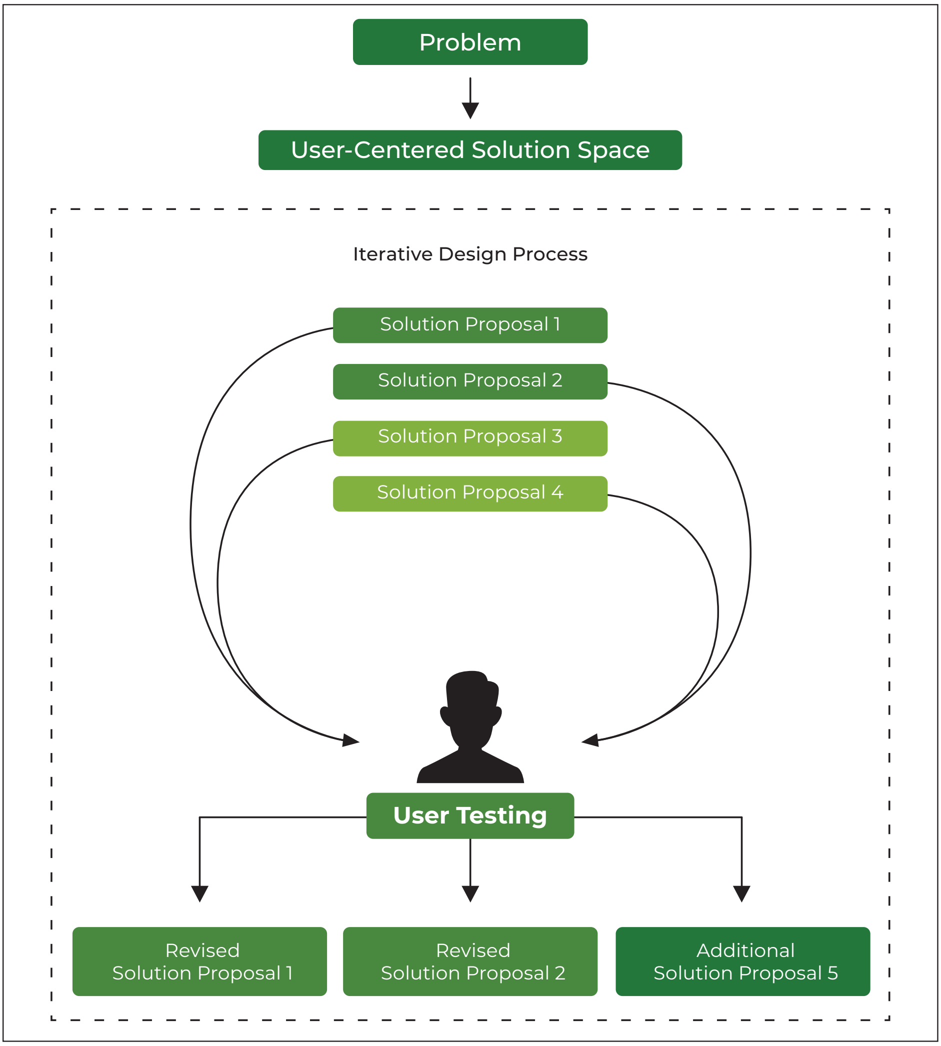

Problem
User-Centered Solution Space
Iterative Design Process
Solution Proposal 1
Solution Proposal 2
Solution Proposal 3
Solution Proposal 4
User Testing
Revised Solution Proposal 1
Revised Solution Proposal 2
Additional Solution Proposal 5

Figure 4.15 Developing a product with a human-centered approach. User testing allows new insights, leading to the exclusion of variants and the development of new ones.

Since the processes described in this chapter consider the customer/user as the final expert on product fit, the processes assume that the design process will be iterative since the initial solution will not be the final solution, as shown in Figure 4.15.

In this context, it is explicitly desired that different solutions be tried and tested. Solutions that fit are retained and further developed; solutions that don't fit are discarded: *When was the last time you started a design with the honest willingness to "throw it away" if it didn't fit?*

4.6.3 Learning from Direct User/Customer Feedback

Designers, psychologists, programmers, engineers, and so on don't make the final decisions; all processes rely on direct customer/user feedback. This feedback can be qualitative feedback gathered through formal tests or via forms. You might have quantitative feedback on purchases, usage, or features used, depending on the stage of development of your solution.

But every single process measures feedback from real users: *How do you measure—objectively and subjectively—whether your design is working?* And by that, we mean beyond the satisfaction of paying customers.

But measuring is not enough. Having numbers and information does not make your solution successful. Only learning from the information enables you to gain a better market position. So, you must use the data to derive improvements. We've met many customers who get lots of feedback from their customers but don't know how to use that feedback to improve their solution systematically: *How do you ensure that you're both a collector and a learner?*

4.6.4 Speed as an Objective

In all processes, ensuring good usability and UX is clearly a core part of development, but only one part. All processes merge into other activities or hand over results to other areas or business areas. To make your usability and UX design as successful as possible, spending as little time as possible on the "wrong" solutions and getting on the "right" path as quickly as possible are essential. This approach will ensure that your time—and the time of everyone else who continues to work with your input—is used in the best possible way.

If you're in a startup, time is the most decisive criterion: Your money will only last for a short time, and at some point, you'll have to earn new money. The sooner you can present a successful product, the better. All processes—except for the medical process—emphasize the importance of fast trials and the need to keep cycle time (i.e., time for each iteration) as short as possible: *How fast are you currently revising your product?*

4.6.5 Success through Collaboration

The core activities described in the processes described in this chapter clearly cannot be performed by one person alone, especially the design, conception, and research phases, as shown in Figure 4.16.

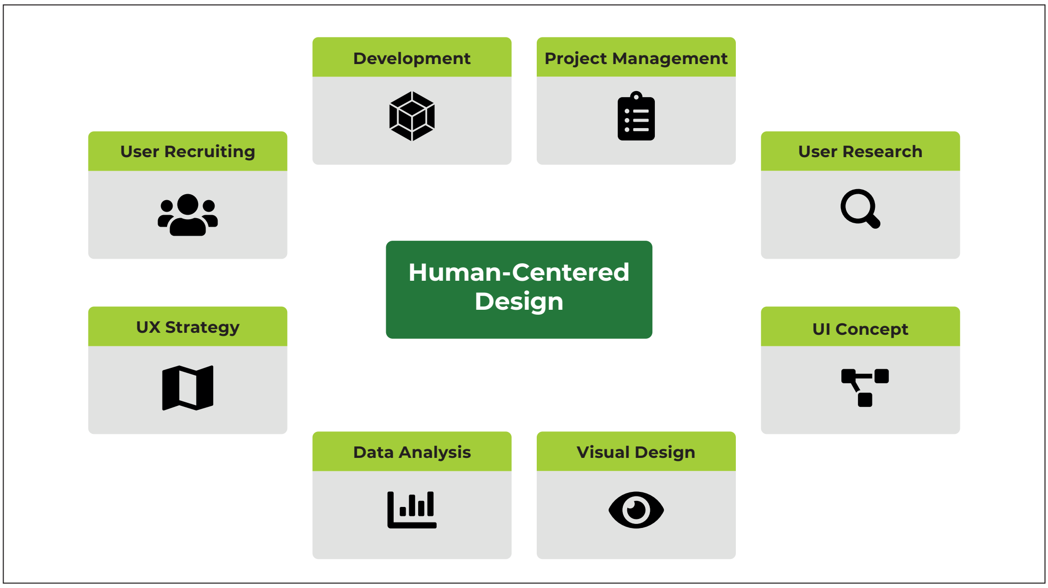


Figure 4.16 Different competencies needed to develop in a human-centered way.

Many different competencies come together. The more complex the problem, the more creative the solution may have to be. Solving complex problems is easier when different perspectives and diverse experience backgrounds are involved: *Which competencies and disciplines are involved in your product?*

4.6.6 Visualization as a Means of Choice

In visual design, you can easily see why results should be pictorially represented. But a pictorial representation is helpful even for intermediate steps, for a summary of the problem, or for early ideas for solutions. Pictures are often easier to grasp and remember, as shown in Figure 4.17. Even if not all processes explicitly mention visualization, all mention some way of making things tangible, at the latest, during the creation of prototypes.

Significantly, when a lot of information is presented, images also help with orientation and ensure that different people can communicate more quickly. This rule applies to all team members, stakeholders, and users working on the project: *How vital are visualizations in your everyday work?*

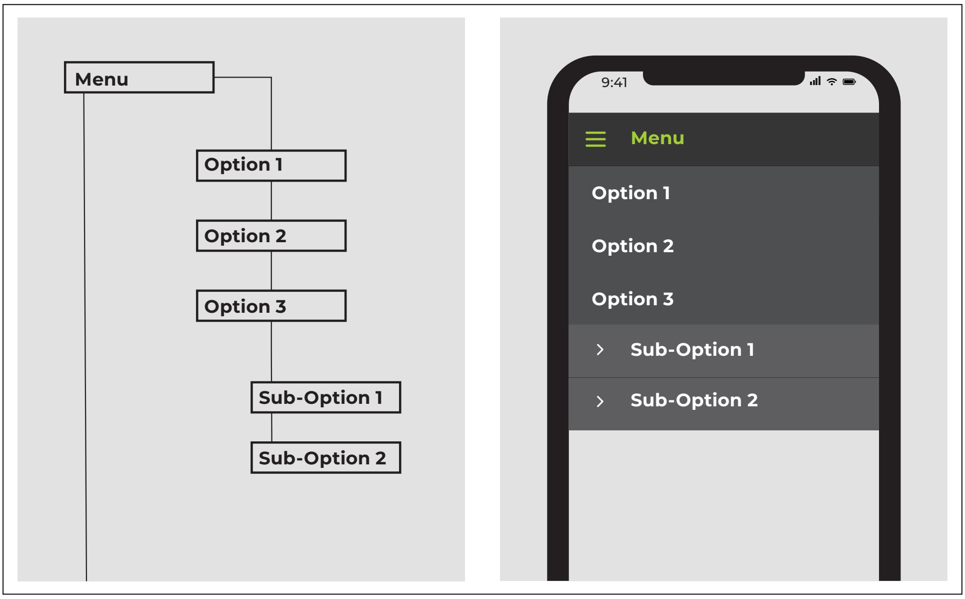

Menu
Option 1
Option 2
Option 3
Sub-Option 1
Sub-Option 2
9:41
Menu
Option 1
Option 2
Option 3
Sub-Option 1
Sub-Option 2

Figure 4.17 Advantages of visualization.

Chapter 5

Data-Driven UX Design

"There is only one boss: the customer. And he can fire everybody in the company, from the chairman on down, simply by spending his money somewhere else."
—Sam Walton (founder of Walmart)

In the previous chapter, we introduced you to various approaches for designing solutions with good usability and user experience (UX). Now, we would like to propose a process that combines the core elements of the processes we presented in Chapter 4 that has worked well for us in many industries in recent years.

In our daily work—and thus also in work for this book—we have decided against simply adopting one of the processes since all processes contain interesting aspects for development. However, the process we follow in the remainder of this book is firmly based on International Organization for Standardization (ISO) 9241-210 and supplements the standard with elements of risk assessment from the medical standard International Electrotechnical Commission (IEC) 62366-1.

We call the process we use for structuring design methods in the course of a project *data-driven UX design (3DUX)*. The name emphasizes that, in our view, a product with good usability and UX can only succeed if actual data from contact with users is integrated. Design processes that dispense with user feedback in all forms at best lead to success through luck, which is not the path we would like to show you.

So first, let's start by presenting an overview of our objective with the data-driven UX design process. We'll explain which logical steps you should go through in terms of time. The methods for each step and a more in-depth look at each step will be provided in Chapter 8 through Chapter 13. Following the brief outline of the process, we'll cover what type and amount of data we would recommend for use in the process, depending on the flexibility and risk of your application.

5.1 The Goal of the Data-Driven UX Design Process

Our goal is to provide a process that empowers you to make informed product and business decisions based on data, not on gut instincts. We also integrate significantly more consideration of the risks associated with product use than other processes outside medical technology typically do.

In our projects, we repeatedly encounter customers who have been desperately fighting a war about which features are needed in their companies for years, whether a particular product would be used by the users or not. Frequently, the discussion in or between the departments drags on for years, paralyzing development or making it slower than expected.

Using the data-driven UX design process, we can regularly generate “aha” moments, as user feedback allows us to turn opinions into a discussion about data. We want to enable these “aha” moments for you too.

The data-driven UX design process provides arguments for justifying your design decisions, thus simplifying the discussion with all decision-makers involved and protecting you from unwelcome consequences. Finally, you can show that you’ve done everything possible to ensure success, as shown in Figure 5.1.

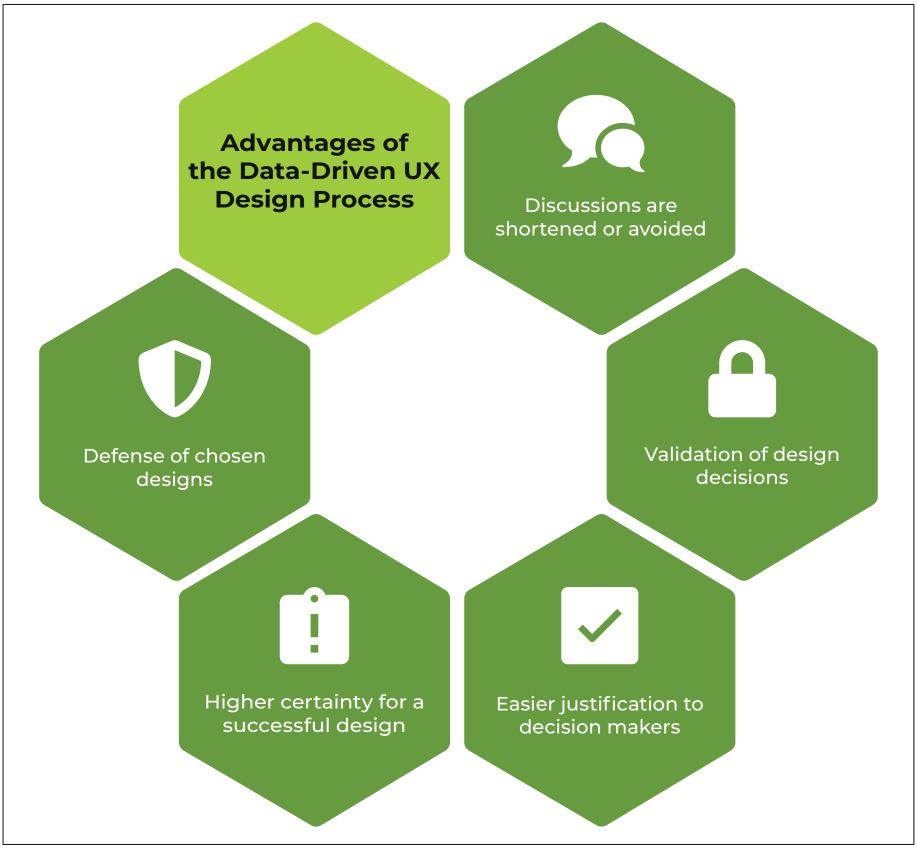

Advantages of the Data-Driven UX Design Process
Discussions are shortened or avoided
Defense of chosen designs
Validation of design decisions
Higher certainty for a successful design
Easier justification to decision makers

Figure 5.1 Advantages of the data-driven UX design process.

In our experience, our process also saves products from unfounded changes at the last moment. If no data is available, we regularly experience how, during discussions with management, features suddenly need to be added or removed, designs must be adapted, or other product decisions must be made at the very last minute based on gut instinct. This phenomenon is also known as the *highest paid person's opinion (HIPPO)* effect. HIPPO is often the deciding factor. Again, the data-driven UX design process guards against HIPPO because you'll have data to justify and defend your design decisions.

Last but not least, our process aims to get you and your team to a successful solution design more quickly and more reliably. But how does data-driven UX design work?

5.2 The Data-Driven UX Design Process

An ideal sequence for the process is shown in Figure 5.2. The sequence starts at the beginning of solution development. Even if you're still considering developing a new system, product, or service, taking the first steps we describe before you develop a technical prototype is a good idea.

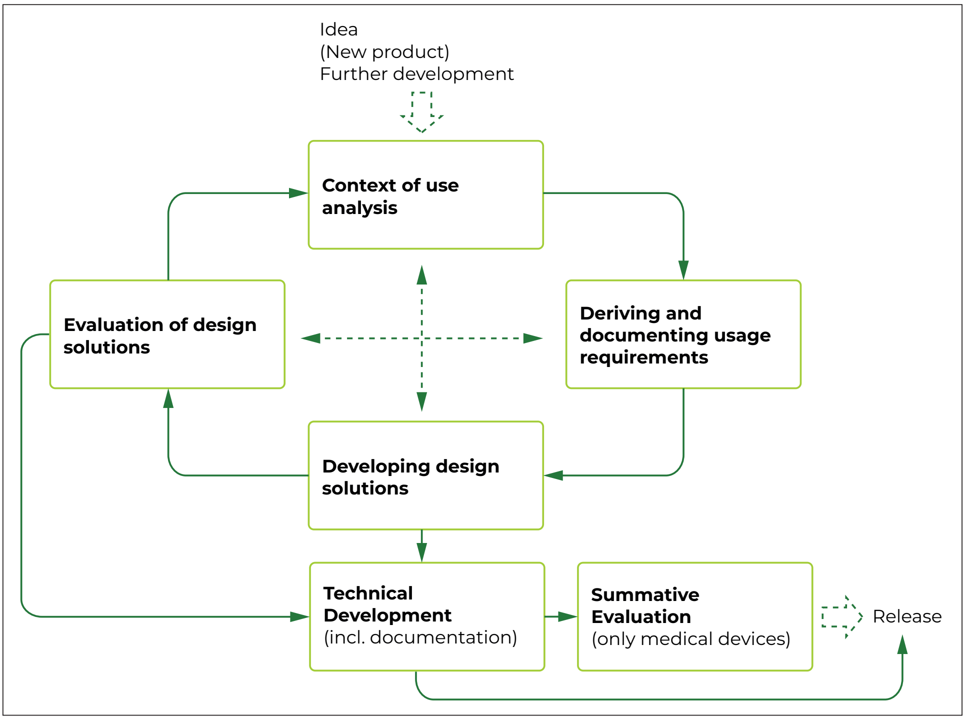

Idea
(New product)
Further development
Context of use analysis
Deriving and documenting usage requirements
Evaluation of design solutions
Developing design solutions
Technical Development
(incl. documentation)
Summative Evaluation
(only medical devices)
Release

Figure 5.2 Overview of the data-driven UX design process.

Few projects start from a "greenfield" scenario; instead, they start with existing products and systems. Therefore, you can, of course, enter the process at any time. From experience, we assure you that you can always select, combine, and apply methods in such a way that they add value to further development—regardless of your current status.

However, these methods add the most value if you start as early as possible and do as little other work as possible beforehand. This efficiency stems from making decisions based on user data in data-driven UX design. If you only integrate this user data into development later, then changes will likely be much more extensive and, therefore, more expensive. Consequently, we'll show you the process from start to finish.

Think of the data-driven UX design process as an approach that makes you a blueprint for your future solution. When you finish the process, you "only" need to implement the solution (e.g., program it). But you have already defined the usability, the UX, and the user acceptance.

You can adjust the process scope depending on your situation; on the system, product, or service; and the competitive environment. In this chapter, we'll explore a tool we call the *flexibility and risk matrix*. This matrix enables informed decision-making about the required data scope and thus controls the scope of the process. But first, let's briefly introduce how the work with our process should be carried out so that it brings the most significant value.

5.2.1 How to Get the Most Value from the Process

To help you get the most out of data-driven UX design, let's recap some preliminary assumptions we've been making:

- **Usability and UX design should be about developing solutions for people.**
  The data-driven UX design process combines the approaches described in Chapter 4 to design systems, products, and services with good usability and UX.

  Thus, using this process only makes sense if you design products with user interfaces (UIs). Any products, systems, or services "under the hood," without direct contact with users, will not benefit from the data-driven UX design process. For all other products, systems, and services, however, the process helps design the technology to adapt to the users and their needs, not the other way around.
- **You achieve the best results in interdisciplinary teams.**
  The design of human-centered solutions requires many different expertise and perspectives. You need expertise in process design, interviewing and testing users, the proper design of studies, and the visual design of applications. In our experience, success is more likely if you have experts in those areas instead of expecting one person to cover all the expertise needed.
- **Human-centered design happens iteratively.**
  Since we develop for people, decisions about the quality of the solution are also made by people. In the case of the data-driven UX design process, decisions are made by users or stakeholders. Even if we have a lot of experience with the target group, we are not members of the target group, so we must question and iteratively improve our solutions through conversations, feedback, and tests.

Human-centered design—and the step-by-step discovery of new requirements—in the data-driven UX design process is iterative. In our experience, you'll be most successful if you accept the iterative nature of the human-centered approach and plan for various iterations from the outset.

We haven't had a good experience putting a lot of effort into contacting users and stakeholders at the beginning of a project and then feeding off it afterward. We instead recommend working fast and keeping the iterations as short as possible. Then, you'll discover missing requirements and open issues more quickly.

- **Decisions are made analytically and based on risk.**
  Decisions within the data-driven UX design process are made analytically and are risk based. This description includes the rigorous derivation of features and their designs based on user and stakeholder feedback and needs. This derivation should always be based on the data you've collected throughout the process. After all, testing and revising a design iteratively has little value if you don't consider user feedback!

  In addition, considering errors and usability problems, in particular, should be analytical. By "analytical," we mean that the cause of the error is traced back, in terms of information processing, to derive suitable design solutions in a targeted manner.

  We recommend that you focus on critical errors and usability problems, then on widespread problems, and then gradually work your way up to the other problems. This approach helps you focus on the most severe issues and makes it easier to present your work in a well-structured and cleanly derived manner. Your results become easier to understand for everyone else, which will ensure you hold all the aces in heated discussions with decision-makers, thus enabling you to make solid arguments. Of course, you can and should garnish all these justifications with the experiences you have already gained.

5.2.2 Brief Overview of the Data-Driven UX Design Process

Our data-driven UX design process consists of four successive, iterative steps, followed by further steps for implementation and development. The basic framework of this process is based on ISO 9241-210, supplemented by risk aspects based on IEC 62366-1. Whenever we speak of "risk" in this book, we mean: *Risk, defined as the probability of the occurrence of an event multiplied by its severity*.

The process starts with a context of use analysis, including risk assessment and documentation, followed by the derivation and documentation of the usage requirements. Subsequently, design solutions are developed from the usage requirements, with prioritization during development based on risk and frequency of use.

The iterative part concludes with the evaluation of the design solutions. The focus of the evaluation is again selected based on risk and frequency of use. Depending on the result, previous steps are repeated, or (technical) development is started. In the case of medical devices, the summative evaluation still follows from a usability point of view before the product is placed on the market. In the following sections, we present the steps of the process in more detail.

5.2.3 Context of Use Analysis, Including Risk Assessment and Documentation

The first step in the data-driven UX design process is the *context of use analysis*, as shown in Figure 5.3. In this activity, we join together all the processes that assume that successful design always starts with understanding the user. Our question is "*What do you think a system, product, or service is*?" Our answer is "*The solution to a problem!*"

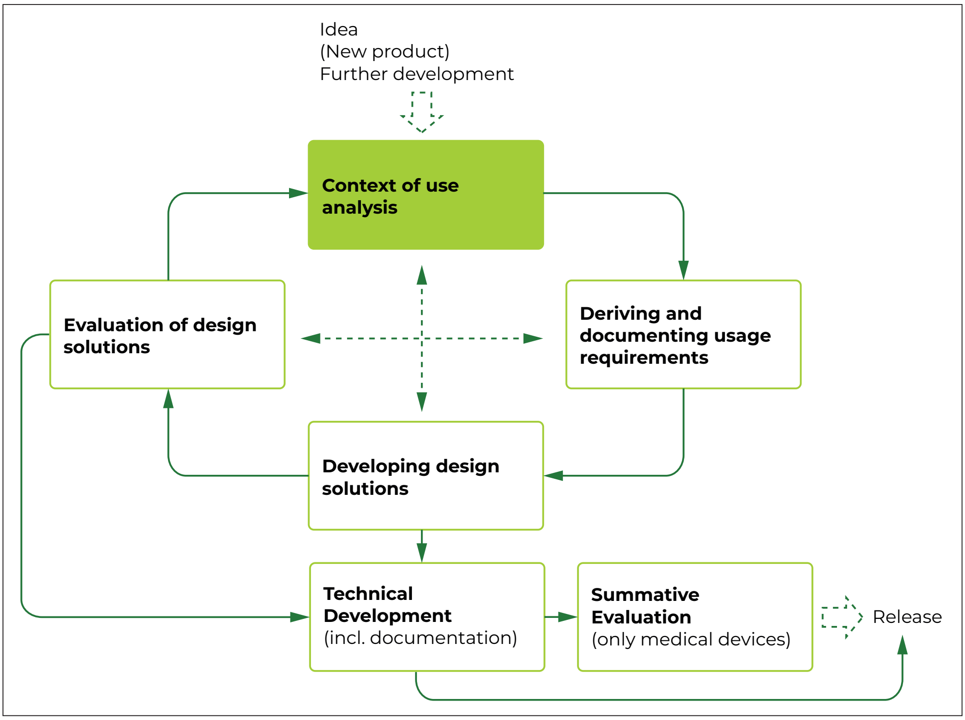

Idea
(New product)
Further development
Context of use analysis
Evaluation of design solutions
Deriving and documenting usage requirements
Developing design solutions
Technical Development (incl. documentation)
Summative Evaluation (only medical devices)
Release

Figure 5.3 Data-driven UX design process: context of use analysis.

No matter what problem you look at in the world, it is always a problem that affects people. Now, you might say, "*Wait a minute, everybody's talking about environmental protection. That's not a people problem.*" Yes, it is. Environmental protection is only perceived as a problem when people suffer from it: because it's too hot, food is scarce, a nice vacation spot is no longer above water, and so on. Before the problem arises, these issues rarely or even never come to mind. As a result, you're constantly solving problems for people, and to do that, you must understand people's problems, as shown in Figure 5.4.

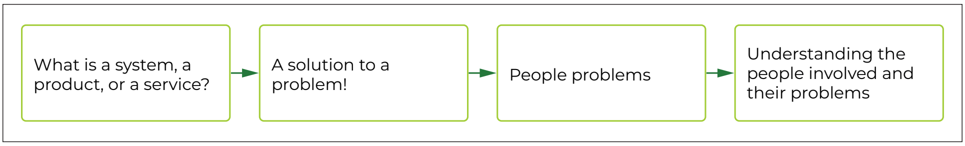

What is a system, a product, or a service?
A solution to a problem!
People problems
Understanding the people involved and their problems

Figure 5.4 To design a system, product, or service, you must understand the people involved and their problems.

Sometimes, you solve people's problems in a similar situation to yours. Then it feels excessive to start the process with a detailed analysis. In many situations, however, you'll find that you have little experience with the environment yourself simply because of the application area.

For example, we specialize in the business-to-business (B2B) environment with complex and critical applications and the medical market. In the last two decades, we've cut down trees, visited steel mills, visited neonatal intensive care units, accompanied waste management workers, designed emergency landing systems for pilots in large aircraft, and observed control rooms running subways, to name just a few application areas. In every single case, we quickly saw that the expert on the environment was not us but the user sitting opposite us.

Believe us when we say: You may feel that you know the target group well or belong to the target group yourself, but you should take the time for a detailed context of use analysis that goes beyond a short brainstorming session in the development team. Therefore, in this step, you'll collect data about the context of use. This data can be existing data (especially if you already have a predecessor product on the market), or this data could be data you're still generating in contact with users.

In our view, the context of use also includes a *risk classification*. In this context, we draw upon the ideas of IEC 62366-1, which considers possible hazards and hazard-related usage scenarios at the beginning of the usability engineering process. You don't necessarily have to create scenarios at this point, but you should consider how risky the application you're working with is. This assessment should lay the foundation for the amount of data you generate during the process. In this context, a higher risk posed by operating errors means you need more data, as shown in Figure 5.5. A low risk associated with mistaken operations implies that you need less data.

Even though few or even no requirements exist for the documentation of findings in most non-regulated areas, we feel that basic documentation is a prerequisite for the data-driven UX design process to show its full strength. To avoid discussions later, you want to substantiate your decisions based on the data, and the documentation comes in handy. The format of the documentation is largely irrelevant; the main thing is that you and your stakeholders still understand later what you wanted to remember.

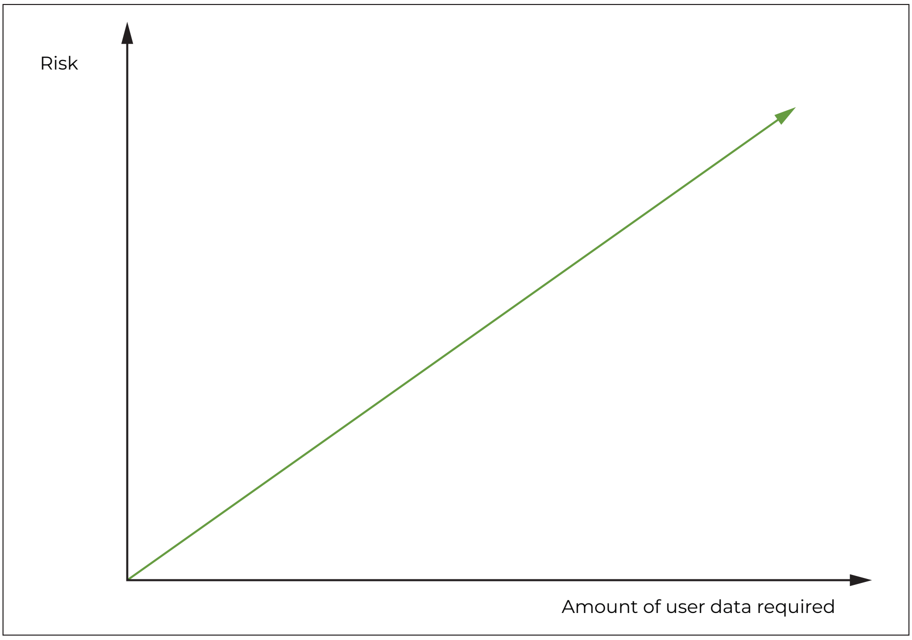


Figure 5.5 The greater the risk posed by your system, product, or service, the greater the amount of (user) data required.

5.2.4 Deriving and Documenting Usage Requirements

In the context of use analysis, you have gotten to know your users and their environment. Hopefully, you have deeply understood the user group and their needs. Now, you're ready to design, right? Not quite yet.

In the spirit of the Double Diamond model and design thinking, we ask you to focus first. Through the context of use analysis, you have received a great deal of new information that now needs to be structured. Structuring and focusing will help you get an overview of the data and choose a weighting that reflects the results of the context of use analysis. Without this structuring and focusing, you risk forgetting much of the information within a short period or—even worse—setting the wrong priorities in development because you weigh all the information equally.

This structuring should be performed by deriving usage requirements for the solution, as shown in Figure 5.6.

These requirements may be vague at the beginning. Still, they become sharper with each iteration and thus enable you to design the appropriate solution step by step in a targeted manner and test the solution approaches found, as shown in Figure 5.7. In addition to the concreteness, the scope of the user requirements will also change during the process. Therefore, requirements not yet foreseeable at the beginning of the development will gradually be added.

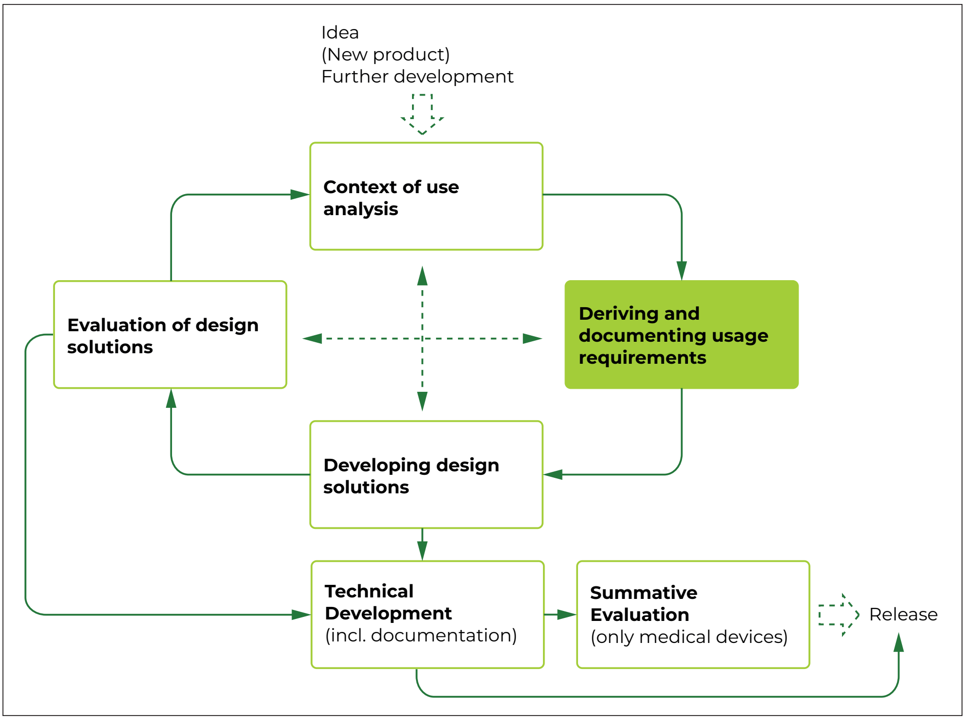

Idea
(New product)
Further development
Context of use analysis
Deriving and documenting usage requirements
Evaluation of design solutions
Developing design solutions
Technical Development (incl. documentation)
Summative Evaluation (only medical devices)
Release

Figure 5.6 Data-driven UX design process: deriving usage requirements.

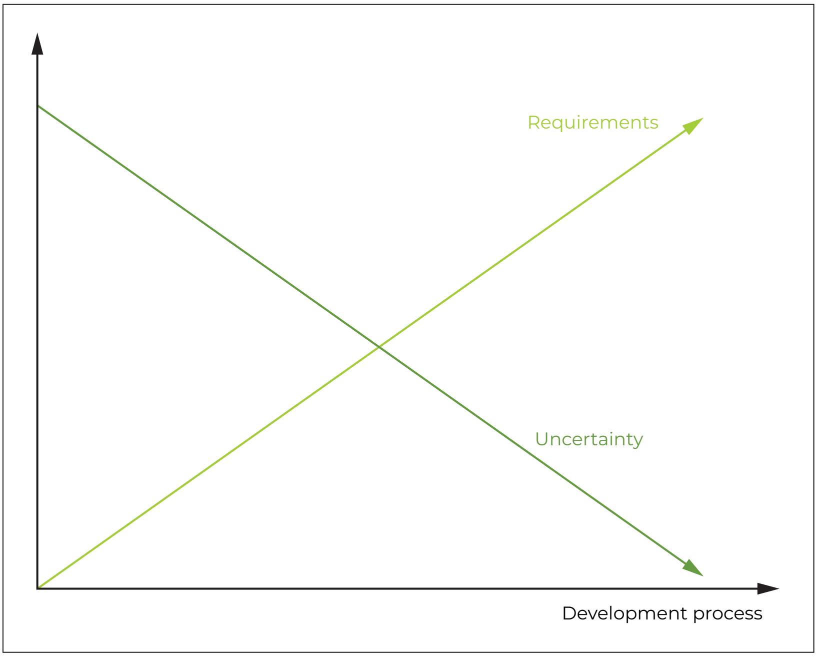


Figure 5.7 The requirements are still vague at the beginning of the development process. Through iterations, they become more precise, and new requirements are added.

The step-by-step addition of new requirements is neither critical nor unusual in an iterative process; it's a normal part of development. Thus, documentation of the requirements is also essential now so that individual aspects are not lost over time and requirements are not forgotten.

5.2.5 Developing Design Solutions Based on Risk and Usage

Once the initial requirements are known, now is the time to develop design solutions, as shown in Figure 5.8. Within the data-driven UX design process, we understand the *development of design solutions* as a fast and prototype-centered process and not technical development. To use the available resources as sparingly as possible, we work with quickly designed prototypes that make it easy for users to imagine how they will interact with the solution. The main goals of this phase are to find the right path and to provide testable prototypes early on.

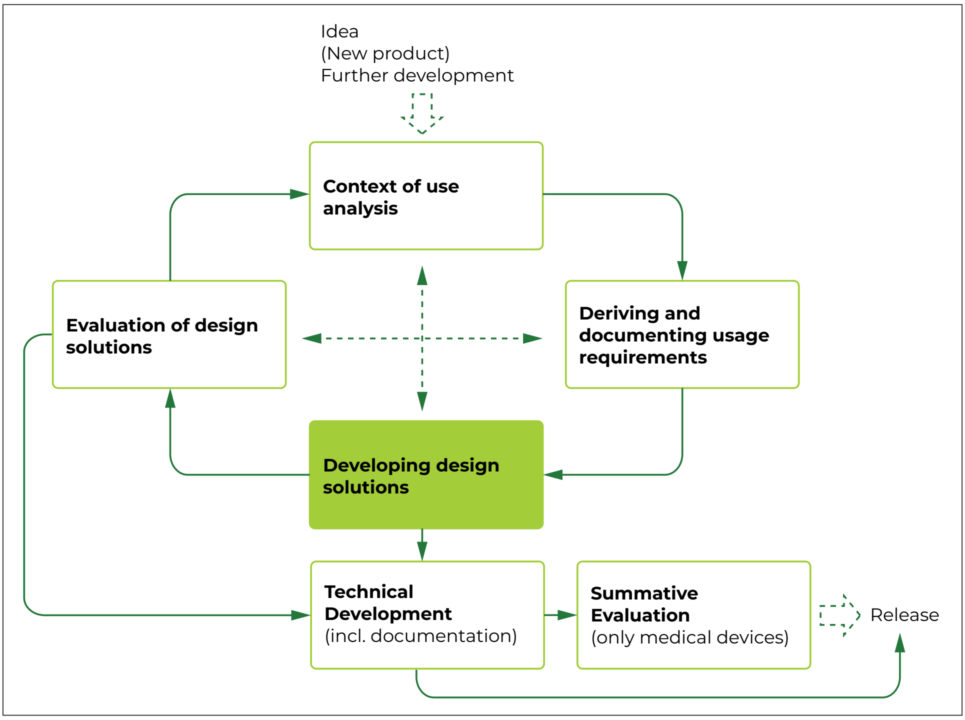

Idea
(New product)
Further development
Context of use analysis
Evaluation of design solutions
Deriving and documenting usage requirements
Developing design solutions
Technical Development
(incl. documentation)
Summative Evaluation
(only medical devices)
Release

Figure 5.8 Data-driven UX design process: developing design solutions.

As described earlier, we assume that the entire process is iterative and that you (just like us) will need several attempts to find an optimal or at least sufficiently good solution. As a result, you'll conduct multiple tests of your designs and then decide which options to pursue and which solutions to discard. We encourage you to proceed based on your risk assessment.

All elements that can lead to hazards during use and where incorrect use poses a risk to users, third parties, or the company should be designed for at the beginning, as shown in Figure 5.9. (In Section 5.3, we'll go into this in more detail.) After accounting for risk, you'll deal with the frequently used elements that are elementary for the solution's success.

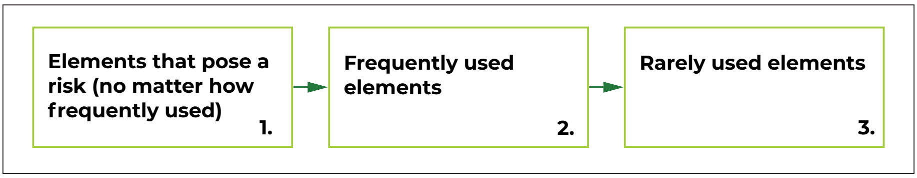


Figure 5.9 Order of designing in the data-driven UX design process.

When we say "elements" in this context, we're referring primarily to workflows mapped across the solution. By designing these workflows early, you ensure that risky and frequently used workflows are tested multiple times and thus go through as many iterations and improvements as possible. With each iteration, you add more workflows and parts of the solution, descending by risk and frequency of use so that the "least important" elements are implemented only at the end and tested infrequently or perhaps not at all. This approach works just as well for apps and software applications as it does for physical products that need to be operated.

If you still have entirely different solutions in the room at the beginning (e.g., a service versus an app versus an app with a combined service), we recommend that you first map and test the idea on a rough level before starting any detailed work. After all, you don't need to define what the workflows will look like in the app if you end up with no app.

Again, we realize that if you work in an app development company, you most likely won't even ask yourself *if* it should be an app. However, this question can be helpful to open up your thinking, come up with alternative solutions, and better classify feedback from users.

We recommend keeping all intermediate results for the later justification of your design solution. In this way, you can tell a coherent and exciting story about the development of your application afterward. This story will make it much easier for you to argue for your design solution and also save you from repeating mistakes you had corrected in advance during later revisions and iterations.

Tip: Reconcile Solutions

Before we jump into the next section, a note on the development of design solutions: Even if we explicitly proceed quickly and prototypically without technical development, the results of this step are, after all, to be implemented later. Only if the concepts in this process step ultimately become a real system, product, or service will your work bear fruit and generate value for users. Therefore, integrating the technical implementation partners as stakeholders directly can help.

We've always had the best experience discussing our designs and design solutions with our clients (both internal and external) *and* our technical implementation partners before user testing. Doing so prevents you from getting good user feedback on solutions that will never see the light of day due to the effort involved or other technical restrictions. Nothing is more frustrating than having designed an excellent UI that no one ever gets to see in a final product.

5.2.6 Evaluation of Design Solutions Prioritized by Risk and Frequency of Use

You might recall our earlier emphasis on the word "analytical" in Section 5.2.1. Now comes another part in which analysis and learning play significant roles, as shown in Figure 5.10.

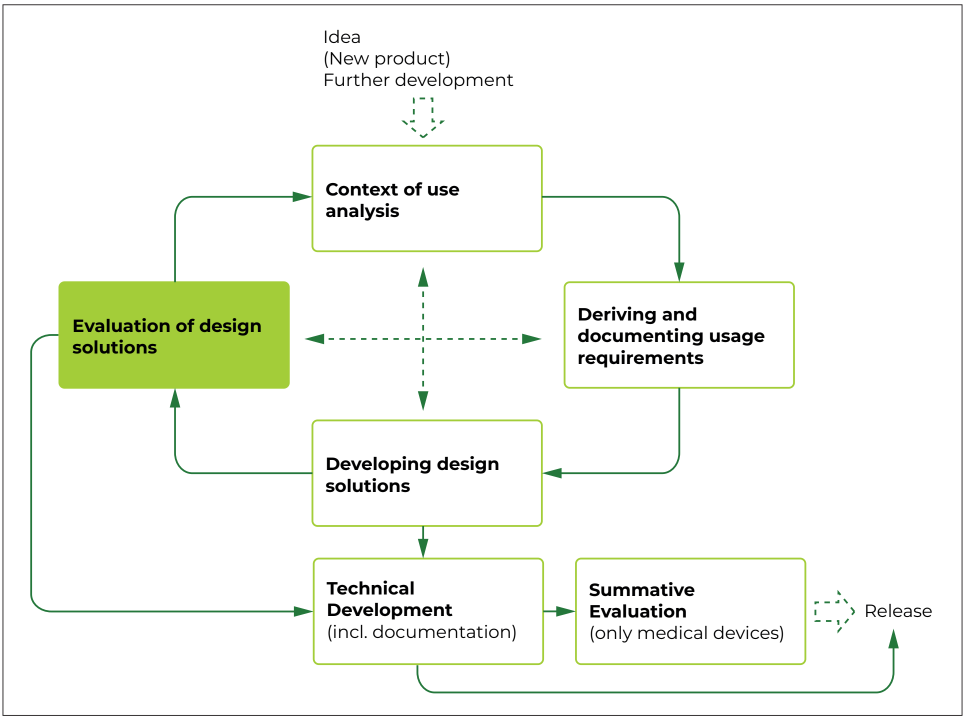

Idea
(New product)
Further development
Context of use analysis
Evaluation of design solutions
Deriving and documenting usage requirements
Developing design solutions
Technical Development
(incl. documentation)
Summative Evaluation
(only medical devices)
Release

Figure 5.10 Data-driven UX design process: evaluation of design solutions.

After you have developed design solutions, you evaluate the solutions. In principle, you have a variety of possible methods that can be used for this evaluation. We'll present the methods in detail later in Chapter 12. However, remember that you'll need feedback from your users—in some form—for a meaningful evaluation of your design solution. Expert evaluations can only replace user evaluations in the rarest of cases.

Similar to the designing step, when evaluating, we prioritize riskier and more frequently used workflows higher than other workflows. Depending on the scope of the solution, you may not have the chance to test *all* aspects *all the* time. Therefore, we test the risky and frequently used workflows as often as possible. In this way, even if you only have a small number of users each time, your most essential workflows will be evaluated by many people. This approach increases your confidence that you won't miss any relevant problems and then be unpleasantly surprised during live operations. How often you test depends again on the overall risk, complexity, prior experience, and ability to make changes after release.

You'll then evaluate the data that you gain from the evaluation. Your goal in this step should be to understand what *works well* and what does not. *Working well* refers to all of the following aspects:

- Users understand the content.
- The workflows are transparent for users to follow.
- Users can go through the tasks independently.
- The design pleases users subjectively.

Now is when the iterative nature of the data-driven UX design process is most evident, as the evaluation results determine how to proceed downstream through the following questions, as shown in Figure 5.11:

- **Has the solution been wholly rejected and deemed not to fit?**
  Discard this solution and look for a new approach. You may need to go as far back as returning to the context of use analysis if the results indicate that you completely missed essential elements. At the very least, however, you'll add individual requirements and revise the design.
- **Was the approach appropriate, but the implementation itself was still not correct?**
  Then, stay with the general direction but refine the implementation. In this case, start more at the design solutions stage and work on the workflows and implementations. You must optimize the parts that provoked negative feedback without worsening the good points. If user input has been comprehensively positive, then move on to implementation.

The implementation can be performed in different ways. Either you submit the complete package for implementation—more or less "throwing it at the wall and seeing what sticks" to borrow an old phrase—or you hand over individual parts that work well to the implementation team, even as you work with your team on the remaining pieces. This scenario is often how collaboration works in agile development environments.

No matter which type of implementation is chosen for your projects, be prepared that there will still be queries and agreements after submission, during which one or two things may be adjusted again. The only important thing is that you don't make any compromises when adapting for implementation that save three days of development time, but mean 5% more call center calls down the road. The data from user tests will help you decide which compromises you can make and which you cannot.

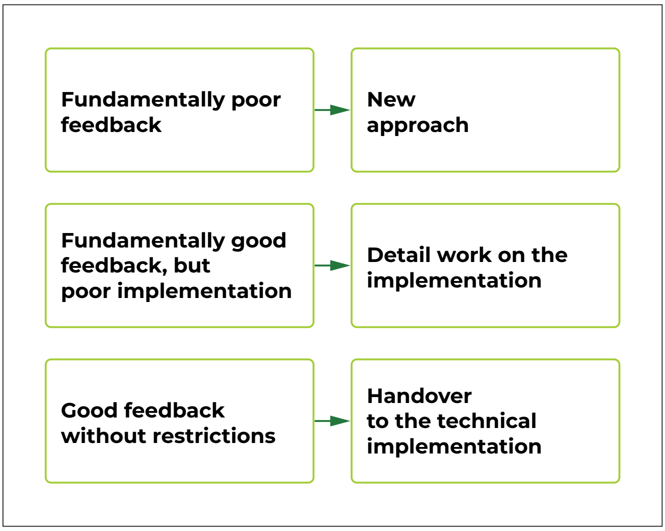

Fundamentally poor feedback
New approach
Fundamentally good feedback, but poor implementation
Detail work on the implementation
Good feedback without restrictions
Handover to the technical implementation

Figure 5.11 Further procedure after an evaluation with users based on the feedback received.

In the medical technology industry, a *summative evaluation* occurs at the end of the development process. In this step, your goal is not to improve the solution but to prove that the operation of your UI is free of unacceptable risks (see Chapter 4, Section 4.5).

In most other cases—perhaps aside from marketing in-house or externally—extensively retesting a finished product has little point unless you plan to use the data to continue learning.

And now you're done! Of course, "done" in digital products always means "done for now." In reality, any user feedback that comes to you during live operations is feedback that can be used to revise the product in the next version. Thus, once you start using it, you'll never get rid of the data-driven UX design process. But: How much of it do you need?

5.3 The Flexibility and Risk Matrix: How Much Data Do You Really Need?

In a perfect world, you would first conduct extensive user research, systematically derive requirements, present initial ideas and designs, and test them until the results say, "Now it's perfect!" Unfortunately, we experience this perfect case relatively rarely in the real world.

In reality, we have deadlines and limited resources, which create the need to keep the design process as large as necessary but as small as possible. If you're like many of us, you probably have difficulty deciding how much is "enough." To help you with this question, we have developed the flexibility and risk matrix shown in Figure 5.12.

The idea behind this matrix is to evaluate your system, product, or service concerning the two dimensions of *flexibility* and *risk* and then draw conclusions about the effort you need to invest in data-driven UX design. This tool allows you to estimate the data required for the rest of the process. First, let's introduce you to the *flexibility* dimension.

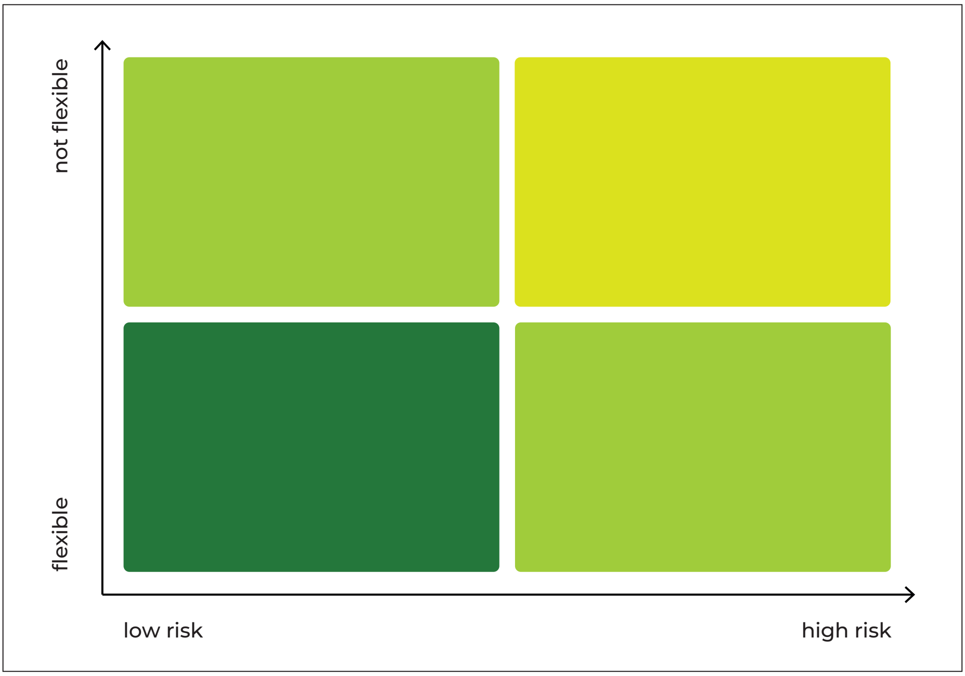


Figure 5.12 Flexibility and risk matrix for determining the required data in the data-driven UX design process.

5.3.1 The Flexibility of Your Solution

By *flexibility*, we don't mean how far you can stretch or bend your product but how adaptable it is after release.

If you manufacture a washing machine, for example, large parts of the product are fixed after development and delivery: The dimensions are defined, the control knobs are fixed, and even the electronics of your device can no longer be changed without great effort. Of course, you could send out a service technician and have all washing machines modified by hand—but the effort involved would most likely be disproportionate to the benefit.

This product therefore has very little flexibility in our perception. In addition, the lifecycle of a washing machine is usually relatively long—about 10 years—which ensures that customers who have just bought a new one will not buy another one for a while. So again, very little flexibility exists in this context. Your washing machine can only be less flexible if you also use the washing machine in a regulated market (let's say, for example, the medical market). Then, in addition to the technical and organizational hurdles to change, regulatory hurdles may even prohibit you from changing the washing machine—at least, without retesting the new design with regulatory agencies. As a result, your product would not be flexible enough to be changed without incurring significant costs and effort for quite some time.

But if your washing machines were "smart," for example, and you, as the manufacturer, could remotely change and adjust various cleaning programs via software updates, your device would gain significantly in flexibility. You still could not change the hardware, but you could change its functionality, at least within certain limits.

Hardware flexibility is also possible in principle. For example, a manufacturer of hot tubs offers the option of exchanging the backrests of the individual seats with a click system. The new backrest creates a unique experience with its different massage jets, which the customer can thus flexibly control himself.

Some complete systems are based entirely on hardware flexibility and have standardized interfaces. Examples include system cameras with exchangeable lenses or computers with exchangeable RAM and slots for add-on cards.

On the other side of the flexibility scale are products that can be easily changed even after market launch. For example, unregulated software products can usually be adapted quickly and flexibly. Manufacturers can make small or far-reaching changes via patches or updates, integrate or remove functions, and even completely replace the UI.

Therefore, the higher the flexibility, the more uncertainty you can accept in development. If it's easy to fix bugs with one click and add features even though the product is already in the field and used, securing the design with data is less relevant. Unless, of course, you're operating in a risky environment.

To determine whether flexibility is more likely to be high or low in your case, ask yourself the following questions:

- How long do your customers and users expect to work with an unchanged version of your solution?
- Can and may your solution be used without prior training of the users?
- Is it possible to update your solution without regulatory or organizational constraints that require (re)testing?

- Does your solution also have internet access during use, which you, as the manufacturer, can access from the outside?
- If no internet access is available, does your product have the ability to have new product versions installed on-site by users?

The more often you answer “yes,” the shorter the period customers and users expect to work with an unchanged product version, and the more flexible your product is. However, if you said “no” frequently, you should spend more time defining user requirements early on and testing them, as your solution will need to be classified as less flexible.

5.3.2 The Risk of Your Solution

The *risk* dimension refers to how dangerous it is if errors occur while using your product, system, or service. Risk in this context is defined according to ISO 14971 in the following formula:

$$\textit{Risk} = \textit{Probability of occurrence} \times \textit{Severity}$$

Therefore, the question is how often this error is likely to occur and how bad its possible consequences result. The risk can relate to different areas:

- The health or life of persons is endangered by operating errors caused by poor design.
- The economic basis of a company is jeopardized by poor design. This could happen, for example, if production losses occur due to faulty operation or if a lawsuit is sought because of these production losses.
- The reputation of a company is jeopardized by poor design. How bad the loss of reputation is for the company must be examined individually based on the brand value and promise.

The easiest way to understand the risk is probably if you’re developing a system, product, or solution whose use is directly related to human health or life.

Perhaps you now have the feeling that this can’t happen to you. However, as UXers, we have worked directly on such projects in many places! Especially in the medical sector, high-risk products are commonplace: developing a heart-lung machine, a defibrillator, or a patient monitor. But even outside the medical industry, we frequently encounter high-risk projects, such as an aircraft emergency landing system or an airport baggage scanner. In all these cases, incorrect operation and misjudgment can result in glaring damage. Therefore, solutions that endanger the health or human life are automatically (high) risk for us.

Also risky, but of course in a different way, are applications that, if misused and misinterpreted by users, cause damage to a company’s economic foundation. This can be your own, but more importantly, it can be your customers' businesses. This category of products includes, for example, process controls in production or software systems that map a company's primary work processes in the productive area. For manufacturing companies, this could be production planning or machine control; for service companies, it could be the project management tool or central data storage.

So, whenever the purpose of the business cannot be fulfilled or can only be partially fulfilled if your system, product, or service is operated incorrectly, we would also rate the application as risky. Compared to the risk of direct harm to people, however, it is more of a medium risk.

The company that runs into economic difficulties due to a poorly designed system, product, or service can also be your own. For example, if you only have the capital for one product development, this last throw should fit and not flop. You could not "afford" a poor design in this case.

The last point—admittedly a relatively soft one, but no less important—is when poor design threatens your company's reputation to such an extent that short-term or lasting damage is to be expected, affecting the perception of the brand and its quality. How severe the damage is, of course, depends heavily on several factors. In our view, these include:

- How established is the brand already?
- What does the brand stand for?
- How closely is the brand linked to the product?

In our view, the potential damage from a poorly designed solution is significantly higher for established brands with outstanding quality than for unknown players.

For example, buying a premium sedan and then struggling with the super-expensive but stubborn-to-operate UI of the infotainment system hurts the brand. It's not uncommon in this case to tell friends or acquaintances that the vehicle itself is excellent, but the operation is horrible. Suddenly, it's not so "premium" anymore.

If you don't have a well-known brand yet, you can't lose brand equity through poor design. But of course, you won't build one either. However, if you work for Apple, for example, and thus even stand for usability, UX, and innovation, the whole thing looks different: The damage caused by a problematic product would be immense.

The same applies if the poorly designed product is closely linked to the customer's brand. Then bad design radiates strongly and clearly onto the core brand—just like good design. An example of products closely related to the core brand are Apple products, such as the Apple Watch. Here, it is clear from the name alone that there is a close connection between the product and the core brand. So, if the Apple Watch is poorly designed, it reflects directly on Apple.

However, the potential damage is lower if your company operates more in the background and presents many different brands to customers. An example of this could be Proctor & Gamble, whose in-house brands include the diaper manufacturer *Pampers* and the cleaning product Swiffer or the chest rubs from Vicks. If a batch of Pampers diapers does not work as intended, it is unlikely that customers will stop buying and using Swiffer.

Of all the factors mentioned, damage to the brand is the hardest to measure. Nevertheless, we would also rate an application as risky here if it is published by a strong brand closely linked to the product and the brand itself stands for high product quality in the premium price segment.

If you want to assess how risky your product, system, or service is, you can use the following questions to make an assessment:

- Can people's lives be endangered by operating your product incorrectly?
- How many lives will be put at risk if there is a mishandling?
- Can incorrect operation of your product threaten the health of people?
- How many people's health is at risk if something goes wrong?
- Can incorrect operation endanger the economic basis of another company or your company?
- Can poor design (not only in terms of usability but also UX) cause lasting damage to the value of your brand?

The more often your answer to these questions is "yes," and the higher the number of people potentially affected, the higher the risk associated with your product, system, or solution. If you could answer "no" to all questions, it is probably a solution with relatively low risk.

5.3.3 Working with the Flexibility and Risk Matrix

With the definition of *flexibility* and *risk* from the previous sections, working with the flexibility and risk matrix is now relatively easy to explain. To work with the matrix, you should ask yourself the following two questions:

1. How flexible is my solution once I have finished developing it and brought it to market?
2. What would be the risk if my solution leads to mistaken operation and misinterpretation?

The point of using the matrix is not to justify your position in the matrix down to a tenth of an inch. Don't fret about whether your solution scores a 58 or a 59 on the flexibility scale. We're just trying to get a sense of the flexibility and risk of your solution.

In general, the further you move your solution to the top right of the matrix, the more data you should obtain during the development of your solution to ensure the quality of your designs, as shown in Figure 5.13. The further you're at the zero point of the matrix at the bottom left, the less effort you need to invest in testing your solution in advance.

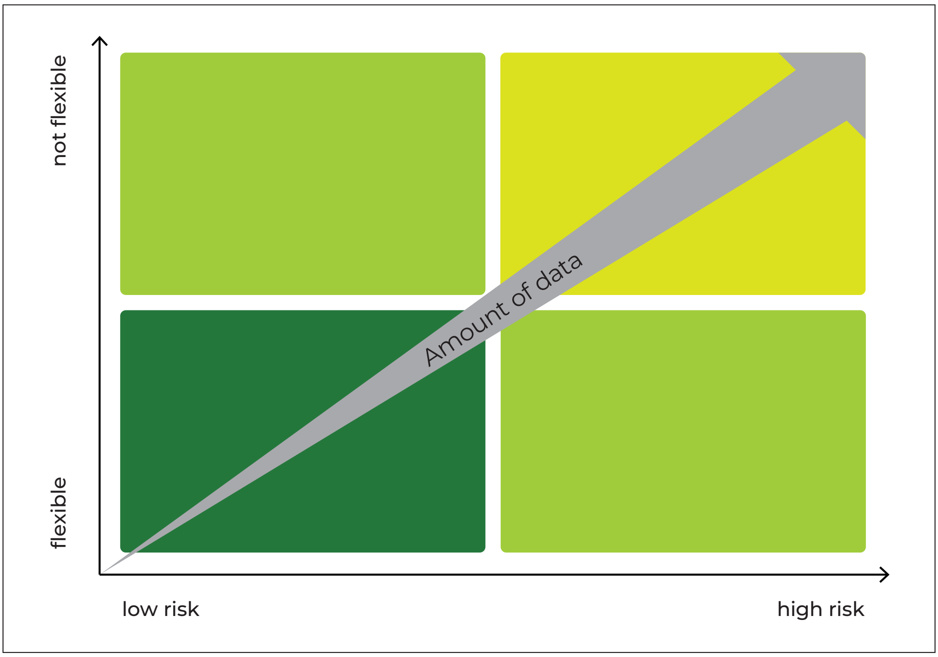


Figure 5.13 The required amount of data according to the flexibility and risk matrix.

The position of your solution on the matrix has several consequences:

- If you have a *low-flex but high-risk application* (top right), you must ensure everything works in advance. In this area, you must invest the most effort, and in the best case, you even perform a final test before release (of course, for a medical device, you must do this anyway, regardless of the matrix). This approach gives you multiple safeguards and also allows you to show that you have invested time and effort to ensure that your product not only looks good in the end but also works safely. An example of a product in this category would be a heart-lung machine: a regulated, approved hardware product that can't be changed once shipped. What's more, people die if this machine is operated incorrectly.
- If you have a *very flexible but potentially risky application* (bottom right), you should also ensure that everything works so that no unnecessary damage occurs. In this area, liability issues and due diligence can suddenly come into play. Again, a final test before release is a good idea. An example product in this area would be Tesla, whose hardware is fixed but can be widely updated via "over the air" software updates. Nevertheless, people can still be harmed in the event of operating errors.

- If you have an *inflexible but low-risk solution* (top left), the worst that happens to you is that you must live permanently with a bad solution. Ultimately, only you (and your customers) can decide whether you want and accept that. In most cases, a final test is unnecessary, and the number of iterations can be reduced compared to risky applications. An example product in this category would be the washing machine from a non-premium manufacturer mentioned earlier. This machine is not flexible due to fixed hardware and software. However, no great (image) damage can be expected as long as the product works.
- If you have a *flexible and low-risk application* (bottom left), you're very comfortable. You can get to market quickly and then address potential issues downstream. From our perspective, you don't need a final test, can test less, or maybe not test at all, and in the best case, you can continue to work with usage data from live operations. An example product would be a to-do app without a well-known brand. The app can be changed flexibly anytime and does no critical damage if errors occur.

You should already have understood how extensive the data-driven UX design process should be for your concrete application. Now, let's focus on the data you'll need.

5.4 Objective versus Subjective Data

In the previous section, we explored if and when you need data. In this section, we'll now deal with the next question: What kind of data is available, and which data do you need? Put simply, two types of data can be obtained from users: *objective data* and *subjective data*.

Subjective data include statements about opinions, attitudes, likes, looks, and so on, which are personal assessments that may differ from person to person and that you cannot infer in advance when viewed from the outside. On the other hand, *objective data* is perceived in the same way by different observers. This kind of data includes, for example, errors in handling a particular UI or operation times. They are, therefore, observable facts.

The distinction is quite simple: Data is objective if you can give the same information to different observers and they all—independently of each other—come to the same conclusion. Data is subjective when this corroboration is not the case. Subjective data usually involves a person's values, attitudes, frameworks, and prior experiences. Subjectivity is a matter of interpreting or evaluating the facts.

For example, you might objectively observe that users make a mistake when using a product. Subjectively, the fact that the user made a mistake may subsequently be perceived as good or bad. If the product is a challenging game of skill, as shown in Figure 5.14, on the left, then one mistake would often also be a great accomplishment that few can manage. If the product is an emergency landing system for pilots, as shown in Figure 5.14, on the right, one mistake would probably be fatal.

Figure 5.14 Skill game (left) versus emergency landing system for pilots (right).

As you have probably noticed from our examples and our discussion of data, subjective and objective data do not necessarily have to be related. Often, data can fundamentally diverge. We often observe the following pattern in UX tests:

1. A new tool is tested for the first time and is not yet good.
2. Users despair (objectively) about every task and cannot solve any of them successfully.
3. In the post-survey, the users give the tool top ratings because it's not the product's fault but their own that it doesn't work.

We do not share this assessment at all. It is never the user's fault if solutions are not self-explanatory and easy to use. It is the fault of the solution. Thus, we urge you to collect objective and subjective data in all tests and evaluations you perform with your system, product, or service. After all, often enough, the exact opposite happens:

1. A tool that has already been designed as a successor to an existing product with a lot of prior knowledge about efficient processes is tested.
2. The users can solve all tasks relatively error free and operate the tool.
3. In the post-survey, simplicity is praised, but users harbor no genuine enthusiasm for the tool.

The objective data speaks for a good tool, and the subjective data indicates something fundamentally wrong. For a comprehensive picture of the current state of your solution, you need both objective and subjective data.

In some areas (e.g., medical technology), objective data is mandatory for product approval. In other industries (e.g., the automotive industry), manufacturers have also committed themselves to comply with objective metrics (e.g., no development of systems where the gaze aversion time from the road exceeds 2 seconds). Note that, if the outcome can be critical, a consensus should exist that the design's success should also be measured objectively. After all, only objective data allows you to analyze the causes of errors and thus systematically eliminate the causes of problems.

In our experience, the type of data you can use changes significantly during development. If you're still at the very beginning of development, then on the one hand, you'll have objective data. However, this data logically does not relate to your solution since it does not yet exist but relates to previous versions or publicly available statistics.

In many cases, when you have little or no data to draw upon, much of the available data will be subjective, as shown in Figure 5.15. You might conduct interviews and focus groups or online surveys that solicit opinions and assessments. Subjective data is often used to understand your target audience better and validate a fundamental design direction.

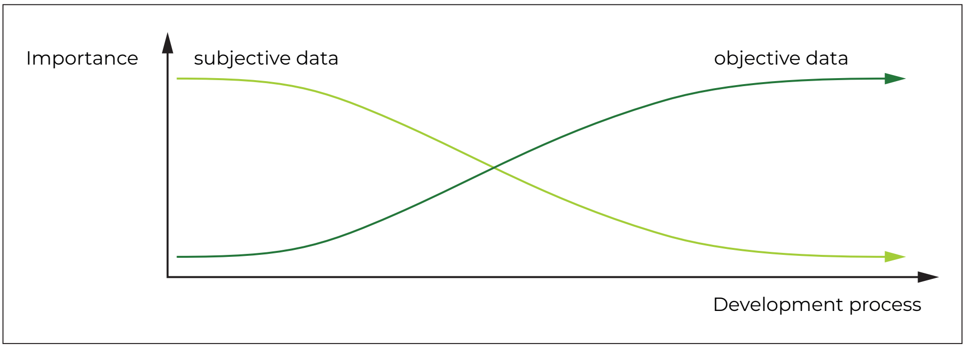


Figure 5.15 Availability and need of subjective and objective data over time of a development.

As development progresses, objective data increasingly supplement subjective data, as shown in Figure 5.15. At the beginning—when you have very rudimentary design solutions—experience shows that you'll only have some objective data, perhaps only in individual dimensions (e.g., error frequency). Nevertheless, you'll still gather a lot of additional information at these points, which will have more of a subjective character, to improve the quality of your design as quickly as possible.

The ratio will suddenly reverse when you're close to completion or even after launch. You'll collect mainly objective data in tests and live operations and selectively supplement this objective data with subjective data. This approach is true until you go back to redevelopment or further development and dive more deeply into user requirements, for which you'll need more subjective data again.

As you can see, whether you need (or can get) more objective or subjective data depends on the time in development and the objective. The more you want to test whether certain parts or processes in your system, product, or service work, the more objective data you need. The more feedback you need on what and how to design, the more subjective data you need.

5.5 Quantitative versus Qualitative Data

In addition to whether you should deal with *objective* or *subjective data*, there is often the discussion of whether quantitative or qualitative data is more needed. The distinction between quantitative and qualitative data can clearly be summarized in the following statements:

- You usually need *qualitative data* to know why and how something happens.
- If you want to know how often something happens, to what extent, or for how long, *quantitative data* is usually the way to go.

Quantitative data can, therefore, typically be represented directly as numerical values. To convert qualitative data into numbers, however, you must first summarize and categorize the data.

Imagine you want to understand in detail your users' problem that you want to solve. In this case, collecting as much qualitative data as possible makes sense. This data could be derived from interviews or observations. The important thing now is that this data helps you understand the situation and helps you learn.

If you can already narrow down the problem, you might be interested in asking, "For how many people is this even a problem?" Then, you could conduct a quantitative online survey to determine the percentage of people with this problem.

Another example of the interplay between quantitative and qualitative data could be the rating of your application on a scale of 5, for example, with the question: "On a scale from '1 = not at all appealing' to '5 = very appealing,' how appealing do you find the application's visual design?"

This question would reflect the quantitative part. As a supplement, we could ask, "What made you give a [here insert the rating your counterpart gave you]? What would we need to do to get one point better?" These responses would be the qualitative portion.

Similar to subjective and objective data, an often helpful approach is to consider both data types together. In general, however, quantitative data comes into play at different times than qualitative data. During development, you could work well with qualitative data in most cases, whereas the final test or data after the release of a product is often quantitative.

So, in general, you can always look at data in both dimensions through the following permutations:

- Objective qualitative data, for example, the type of errors made when using specific software
- Objective quantitative data, for example, the average length of time users visit a website

- Subjective qualitative data, for example, full-text responses to interview questions
- Subjective quantitative data, for example, scale ratings on the simplicity of specific software in an online questionnaire

What kind of data you need (or even have available) always depends on your situation and question.

One point to note: You can also ask for scale values in a study with 5 participants. However, “quantitative” usually means you need a certain amount of data to make meaningful statements. This data usually only makes sense to give a mean value if the data are normally distributed, thus allowing the calculation of a mean value. Typically, having at least 30 data sets is sufficient to ensure a normal distribution, but of course, more data can be used.

For more information on the distinction between objective/subjective and quantitative/qualitative data, we recommend *Statistics: The Art and Science of Learning from Data* (Agresti, Franklin, & Klingenberg, 2020). This book is also suitable for determining how many people you need to make valid quantitative statements.

5.6 Critically Evaluating Data

Now, we would like to go into more detail about how you should handle all this data and how you can make the best possible use of your data. An essential element is critically questioning the data you have. Depending on how you present or view data, you can come to completely different conclusions with the same data. Thus, you must know of the typical pitfalls when working with data.

Especially if you work with quantitative data, many additional pitfalls cannot be listed in this book. If this topic interests you, many textbooks in the field of market research go into much more depth (e.g., Agresti, Franklin, & Klingenberg, 2020; Patten & Newhart, 2017; Hague, 2021).

5.6.1 Where Does the Data Come From?

Data is valuable for designing UIs and products. However, the value of that data is closely related to whether you have collected data from the right people. If admins install an internal B2B application individually on users’ computers, the data will almost inevitably come from the right people.

But when you create an online questionnaire, you usually don’t know whether the people answering the questionnaire belong to the user group from which you want to collect data. Whenever you don’t “see” the people and only see the data submitted, we recommend integrating questions that allow you to filter out data from inappropriate people downstream. Questions could include, for example, age, gender, occupation, or willingness to use in a particular area. However, exactly what questions these are depends, of course, on which user group you want to reach.

Tip: Open Questions as a Filter

Note that participation in studies, tests, surveys, and so on is often associated with an expense allowance. In other words, participants receive money, vouchers, or products for taking part. As a result, people may try to belong to the user group even if they don't and cheat their way into your survey.

We prevent this issue by making the questions for including or excluding participants very open-ended. Thus, for the responder, the "correct" answer to that question is unclear.

Many online tools can provide people with one-time links or cancel a survey if the participation conditions are unmet. Whether this filtering out is necessary is up to you to decide. You can also use the flexibility and risk matrix as a guide: The more risky/inflexible your solution is, the better the data quality you need.

5.6.2 Objective Data or Interpretation

Especially with qualitative data, a common pitfall is for data not to be recorded objectively but instead to be interpreted as early as the documentation stage. For example, an interview with a user might not be recorded with a camera (objective). Instead, the interviewer's interpretation of what was said might be recorded in bullet points (somewhat subjective).

We advise you against the subjective recording of data first. Try to make sure that you first collect the data objectively, perhaps live by taking notes or with recordings. Then, when you have completed the data collection and want to take stock, you can look for patterns in your objective data. Recording data objectively does not necessarily mean that you transcribe everything that was said verbatim. We want to emphasize that you should note the essence of an observation and the essence of what was said, not your interpretation of it.

One great advantage is that you can view and interpret the objective data several times. If necessary, you can also re-analyze the recorded objective data later with more knowledge. If you did not record the objective data in the first place, but instead interpreted it directly, then you lack this option. Because the data interpretation depends on the person doing the interpreting, you lose information. The assessment could have been different for different people. Thus, you might find that, if several people are involved in the data collection, they interpret the same event differently, causing you to miss patterns that should have been obvious from the objective data.

This point is less critical if you're the only person collecting data and working with it. But even then, collecting objective data and only interpreting it later is helpful. This approach also makes it easier to defend your conclusions if necessary.

5.6.3 Is the Frame of Reference Correct?

People's evaluation of a system, product, or service depends on their *frame of reference*. People will still make comparisons if you don't set a frame of reference, but they won't know what they are comparing. Don't believe that? Let's explain.

At this point, we'd like to share a personal story that happened to Michaela in her diploma thesis (equivalent to a master's thesis). During her studies, Michaela specialized in the field of usability and UX, and she worked on whether an object's aesthetics influenced its perceived and actual usability. (Spoiler alert: No.) At least not as far as the experiment could show. But do you know what the problem is? We still don't even know if aesthetics matters. And now we come to the issue of the *frame of reference*.

A Personal Example: The Can Opener Study

In Michaela's thesis, the usability of several can openers was compared. For this purpose, the same can opener was purchased and redesigned several times. The redesign did not change the item's usability, but the visual design changes were severe: There were some hideous things at the end and some quite fancy. To avoid making this classification based on Michaela's assessment, a pre-test was conducted in which the aesthetics of these can openers were compared and evaluated. Then, for the experiment, we could select three can openers perceived completely differently: one neutral, one very beautiful, and one very ugly.

In the actual study, the participants were then divided into three groups. One group was given the neutral can opener; one group, the beautiful can opener; and one group, the ugly can opener. The subjects were then asked to open several cans quickly and accurately and assess each can opener on usability and aesthetics. It turned out that neither the objective usability (i.e., speed and accuracy in this case) nor the subjective usability differed between the three groups.

So far, so good. Unfortunately, however, it also turned out that the assessments of the can openers' aesthetics did not differ between the groups. Thus, the three groups all rated the can openers as equally aesthetic. The problem: Since each group only saw its own can opener (and not the others' can openers), each group had its own frame of reference. The people from the group with the beautiful can opener concluded that even more beautiful can openers must exist. The individuals with the ugly can openers probably figured even uglier can openers exist. To this day, we do not know whether beauty plays a role in the perceived usability of can openers. Unfortunately, finding out with the help of this chosen experimental setup was impossible. Well, you don't make the same mistake twice.

You can see from this can opener story: If you do not provide the frame of reference—for example, by offering comparative products, alternative solutions, previous versions, or variants—then your participants will set their own frames of reference. Therefore, many subjective assessments become less valuable as a result.

5.6.4 Have You Had Value Judgments or Statements Explained?

In interview situations, sometimes, a statement is made that is interpreted entirely differently by the interviewer than by the interviewee. For this reason, as an interviewer, you should ideally inquire about value judgments to record the correct interpretation.

A Personal Example: The Car Mechanic's Car

In an interview on the topic of "vehicle use," Benjamin received a statement from an interviewee that his vehicle is often defective. The interviewee said literally, "*Oh, my car is always broken. There's always something to be done or something to be replaced!*"

How would you interpret this statement? If you were like Benjamin, you would think to yourself at that moment that this must be terrible. After all, a car is for driving, and no one needs a defective vehicle.

Fortunately, Benjamin followed up and asked, "*And what do you think of that?*" because Benjamin's interpretation would have been entirely wrong. It turned out the interviewee loved that the vehicle was permanently broken. He was an enthusiastic hobby mechanic who rode his bicycle to work. He had the car only to tinker with it forever. The interviewee would, therefore, not have needed a functioning vehicle without defects.

Perhaps you can recall this story the next time you hear a statement or evaluation as an interviewer. Ask specific questions!

5.6.5 Is the Data Complete and Meaningful?

When you work with quantitative data (e.g., questionnaire data), the question of data quality is crucial. People rarely want to fill out lengthy questionnaires. The longer and more cumbersome a survey is, the more likely you'll lose people entirely along the way. Sometime, long questionnaires are so demotivating that, although people put in checkmarks, they no longer make sense.

The first check you should make in your data is whether all the questions have been answered. You may have 150 people answering the first question but only 75 getting to the end. This self-filtering of respondents is not bad if you know dropping out of the survey happens "by chance," not because a certain category of people drops. Otherwise, you may end up with only a specific type of person in the survey who will give different answers. Depending on how important this aspect is to you, you can exclude people from the evaluation who have not completed the survey.

The second thing you should check is whether any strange response patterns appear, which is harder to check manually in the digital age. In psychology studies, when you work with paper questionnaires, you should check whether the questionnaires are filled out according to a specific pattern (e.g., in an X-shape, always option 1, or always option 5). Digitally, this pattern matching is less obvious because you don't see the completed questionnaire in front of you; however, you can recognize many patterns from the numbers. Usually, all persons who answer everything with the same answer option should be excluded.

Now, you might say, "Hey, what if the person just loved everything?" Then, we would tell you, "You have a problem with your questionnaire design." Usually, you design questionnaires so that, even if someone thinks everything is great, the questions require different values to be checked off. You achieve this goal by turning some questions around. (Instead of "I found it very easy to use," it then says, "I found it very hard to use.") In this way, a checkmark in the same place becomes a contradictory statement. Additionally, in the case of essential aspects, sometimes very similar questions are asked. Later in the questionnaire evaluation, whether those answers match can be verified.

Clean questionnaire construction is an art in itself. To learn more about this topic, we recommend literature on market research or test and questionnaire construction. But quite frankly, in most cases, you'll be okay with a less complex survey. The most important thing is to ensure the questions are straightforward and that they measure what you want.

5.7 Quick Help for Categorization

In this section, we would like to conclude this chapter with summarized guidance on deciding if, when, and how much data you should use in your development process. We've structured this quiz so that you can either agree or disagree with each prompt. Keep track of the number of "yes" and "no" answers.

5.7.1 Do You Need Data during Development?

Let's summarize again the questions that will help you determine if you even need data during development:

1. If the solution you want to develop is a risky application (for the definition, see Section 5.3.2)?
   - *Yes*: In any case, you need data. A final quality check is also recommended for applications with a high risk. Additionally, check the flexibility.
   - *No*: Check the flexibility of your application.

2. Does the solution you develop have low flexibility (for the definition, see Section 5.3.1) so that changes after completion are impossible, difficult, or expensive to correct?
   - *Yes*: In any case, you need data. In addition, a final quality check is recommended. If you have said "yes" twice, you should try to test your development intensively with users and thus secure it early on.
   - *No*: Is your application not particularly risky but flexible? Then, you can keep the amount of data relatively low during development and learn in live operation. Is your application risky but flexible? Then, you need data and a final quality check—at least in risk-associated areas. If you have low-risk workflows, you can hedge them with less data and gather feedback during live operations.

5.7.2 How Much Data Do You Need?

This section is only relevant if you already know that you need data and have answered "yes" at least once in the previous section.

Now is the time to ask, "How much data do you need?" Let's start round 2 of our test:

- *Not one "yes"*: If you didn't say "yes" at all in the previous section, then this section is not relevant to you.
- *Once "yes"*: You should test at least once during development. If this test looks good, then you're on the right track. You can learn a lot from the test, and if your system, product, or service does not perform well, test again. Testing after development is complete also makes sense because if your solution is risky or inflexible, you don't want to find out something is bad when you get to market.
- *Two times "yes"*: You probably already believe that you should test several times during development. Testing repeatedly will ensure that you're always on the right track. The number of test iterations increases with the risk of your solution. The higher the risk of your solution, the more often you should test it during development. Again, the final test makes sense because you even have "twice" the chance to look bad on the market.

5.7.3 What Type of Data Do You Need?

The final question to ask is the question of the type of data. You can approach this question in two ways: once by asking what data is likely readily available and once by asking what data you need.

The earlier you are in the development process, the more you need qualitative data, probably subjective data in most cases. However, if available, you can use objective data. The later you are in the development process, the more likely you'll have qualitative/objective data. After development, you'll most likely have quantitative/objective data, sometimes combined with quantitative/subjective data from surveys.

On the subject of what data you need, you can use the following questions:

1. Do you want to clarify why and how something is done, used, or not used?
   - *Yes*: You need qualitative data.
   - *No*: Go to the next question.
2. Would you like to know how many, how often, or how long something is done, used, or not used?
   - *Yes*: You need quantitative data.
   - *No*: We are not clear what you want to find out. Write to us at *handbook3DUX@custom-interactions.com*, and we'll help you find a solution.

5.7.4 Conclusion

We hope we've clarified why you need user data to develop great solutions. Whether you need data during development and to what extent depends on the factors we've mentioned throughout this chapter. Fundamentally, we don't believe you can develop entirely without data.

If you don't gather data at the beginning, then when the product is live, you'll get data from your users showing where room for improvement exists. This feedback could be in the form of feature requests, negative reviews, user avoidance, usage behavior data, poor retention rates (the portion of your existing customers who remain your customers after a certain amount of time), and support requests. Countless other data points are available to see if everything fits or where things are stuck in live operations. But believe us when we tell you: The data will find you; when it does, ideally, it's "good" data.

A not-so-nice example of this feedback comes from the video game industry, where user tests (called alpha and beta tests) are conducted regularly. In these tests, players try out the game before its release and provide feedback. In this context, data collection is integrated into the process. However, not all manufacturers take user feedback seriously. One of the biggest publishers—Ubisoft—caused a controversy because, although beta tests were conducted for one of its top games (*Tom Clancy's Ghost Recon Breakpoint*) and all the players' criticisms were available in the forums, there was not an adequate response to the feedback. When the game was released, most technical bugs had been fixed, but feedback about the game's mechanics and lack of gameplay did not appear to have been considered. The consequences included worldwide backlash, bad ratings in gaming reviews, and lost players. A problem for a game development in which about 3,000 developers were involved and whose costs are estimated to be in the hundreds of millions!

So, our conclusion is this: You'll get the feedback of your customers and users—one way or another. Through the targeted use of a data-driven process, you can only decide at what point you want to receive this feedback and when you want to learn from it. In most cases, learning sooner rather than later probably makes more sense. However, if you have learned a lot in advance (e.g., through previous versions or customer feedback analysis), you can rely on your existing data and see what happens.

Chapter 6

A Helpful Mindset for Usability and UX Design

"It's not just about understanding the words. It's also about understanding the person."
—Erich Brandl (musician, author, and lyricist)

Finding this chapter in a usability and user experience (UX) design book may surprise you. But we believe that one's mindset has the most significant influence on the success or failure of one's work. By *mindset*, we mean the attitudes and expectations with which a person approaches a subject. To clarify why mindset is so important, let's take a closer look in this chapter.

Now, ask yourself honestly: Do you think you can learn anything you want to? Or are you of the opinion that you're either good at something or not? Both attitudes affect your life.

In the first case, according to Carol Dweck, an American psychologist who has researched mindset, you currently have more of a *growth mindset*. You can grow from challenges and improve. In the second case, you have more of a *fixed mindset*. You're currently more afraid of a challenge, especially failure. We have deliberately written "currently" because a key point of Dweck's work is that mindsets can change. Additionally, in some areas of life, one type of mindset prevails but in other areas, a different mindset (Dweck 2006; Heckausen & Dweck 1998).

We focus primarily on the professional side of life in this book. When working on usability and UX design topics, users (and thus people) are the focus of interest. When you converse with these users, your mindset decides largely what they take in, believe, and later use. If you currently have a fixed mindset in this area, you may even avoid talking to a user altogether. They could tell you your product is not great and urgently needs improvement. You would perceive this statement as a failure and rather avoid the conversation.

On the other hand, with a growth mindset, you would gratefully accept user feedback and learn what you can improve about your product. The result will be a better product only in the case of the growth mindset. Your mindset plays a crucial role in usability and UX!

In this chapter, we'll cover some stumbling blocks in mindset that we encounter repeatedly in our daily work. Perhaps you'll recognize some of these issues in yourself. That's good because this realization enables you to work on them directly. We want to help you develop and strengthen a growth mindset—the fundamental prerequisite for usability and UX. Let's look at the difference between *assumptions* and *knowledge* to get started.

6.1 Assumptions versus Knowledge

One of the biggest problems we encounter in our work is confusion between *assumptions* and *knowledge*. An *assumption* is a guess or opinion taken for granted, while *knowledge* is the "fact or condition of knowing something with familiarity gained through experience or association" (*https://www.merriam-webster.com/dictionary/knowledge*). When you make an assumption about something, you guess something is so but have no evidence to support that claim. If, on the contrary, you know something, then you can prove that your claim is based on facts, in highly simplified terms (see also Chapter 5, Section 5.6).

We constantly make assumptions in our lives, typically causing no problems or just minor difficulties but occasionally major ones. For example, a teacher snaps at the class clown because someone put a wet sponge on her chair when her back was turned. In this case, the teacher assumed the culprit *must* be the same as always: the class clown. At this point, we don't know if the class clown actually was the culprit. Nevertheless, our assumptions can trigger reactions that lead us to snap at one person in particular.

Although assumptions can lead to difficulties, they are still essential for all of us in our daily lives. In their book, Sahakian and LaBuzetta (2013) write that 35,000 decisions are made daily. Whether we want to get up when the alarm clock rings, what we eat for breakfast, what we wear: Assumptions help us make quick decisions about little things and not go crazy. However, when important decisions need to be made, we recommend that you make decisions based on knowledge. Decisions that affect the success or failure of your product are such important decisions. But how do you know if you know something or only assume it?

6.1.1 Clearly Revealing One's Own Assumptions

Revealing own assumptions is a challenging task. All our assumptions exist in our heads and often seem accurate, so the moment rarely comes when one's assumptions are questioned. So, the first big step is to become aware of your assumptions in general but also in the development of products. Otherwise, you risk building systematic errors into your product based on false assumptions and not noticing them until later, for example, when the product is poorly received by the market.

A simple yet effective way to uncover your assumptions is to work with questions. Ask yourself, "What must be true for my product to succeed?" Then, as a first step, list all the points that come to mind—preferably in an interdisciplinary workshop since each discipline brings unique prior experiences and perspectives.

For example, let's answer this question for Healthyfy—our health platform. Ask yourself, "What must be true for Healthyfy to be a success?" You'll probably come up with various answers like the following answers we came up with spontaneously:

- Patients have an interest in searching for a doctor online.
- Physicians can present their qualifications online.
- Patients can name the specialist they need for their symptoms.
- Reviews from other patients are an appropriate basis for evaluating physicians.
- Physicians have an interest in communicating with patients online.
- Physicians have an interest in screening patients online.
- A physician can perform a decent examination via video conference.
- Physicians have the skills to work with modern telecommunications media.
- Sharing health data through a digital platform is allowed.

These assumptions are just a few examples behind the founding of Healthyfy. If you ask these questions in the context of an interdisciplinary workshop, you may be surprised by the different answers you get.

Once you know the assumptions about your product, you can move on to the next step: identifying knowledge gaps.

6.1.2 Distinguishing Assumptions from Knowledge

Now comes the exciting moment: Together with your interdisciplinary team (ideally) or alone, you have uncovered the assumptions that must be true for your product's success. The next step is to decide whether this information is provable knowledge or a still unprovable assumption.

Our procedure is straightforward and consistently successful: In a workshop, we ask our customers to sort each assumption into one of three color categories. This task is initially performed by the workshop participants alone. The colors and their meanings are specified in the following ways:

- Green stands for "I know because...."
- Yellow stands for "I am relatively sure because...."
- Red stands for "I have no evidence for this."

For a broad overview, we usually draw a simple matrix on a whiteboard for this sorting task, as shown in Figure 6.1. In the matrix, the three colors (green, yellow, and red) form the header. The sorting of the assumptions is then performed using sticky notes, where each sticky note stands for one assumption.

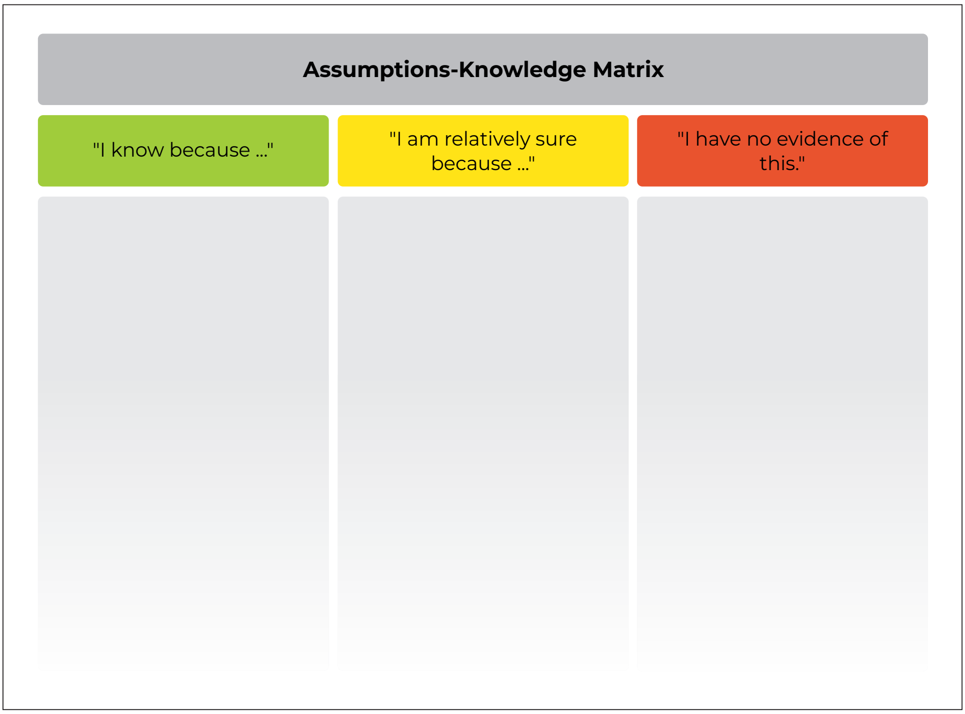

Assumptions-Knowledge Matrix
"I know because ..."
"I am relatively sure because ..."
"I have no evidence of this."

Figure 6.1 Our assumptions-knowledge matrix.

In addition to the pure sorting of assumptions into the three color categories, all participants have the additional task of indicating the sources of their knowledge on the sticky note for the green and yellow cards. In this context, “sources” is to be understood broadly and can refer to scientific sources (e.g., literature and studies), statistics, internal market research results, customer surveys, interview results, competitor results, and more. If possible, the sticky notes in the red category should also be described. Whenever someone has collected sources that speak against an assumption, these sources are noted accordingly, as shown in Figure 6.2.

When selecting sources, they must be verifiable with figures, data, or facts. After all, the point is to separate assumptions from knowledge and to separate gut feelings from facts.

Depending on the number of assumptions identified, this workshop can take a little time. We’ve calculated about 5 minutes per assumption. Thus, if you have identified 12 assumptions, give yourself and the team 60 minutes to go through all the assumptions individually. If you’re quicker and need less time, you can stop anytime. If you need longer, we recommend investing that time anyway.

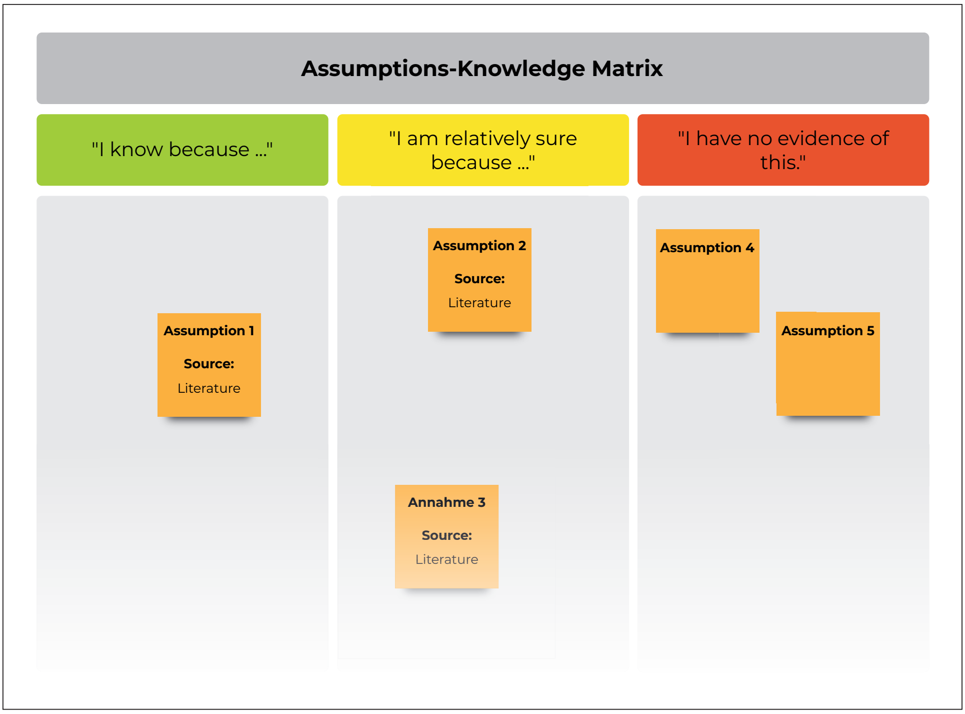

Assumptions-Knowledge Matrix
"I know because ..."
"I am relatively sure because ..."
"I have no evidence of this."
Assumption 2
Source:
Literature
Assumption 4
Assumption 1
Source:
Literature
Assumption 5
Annahme 3
Source:
Literature

Figure 6.2 Completed assumptions-knowledge matrix.

Through this workshop, you can accomplish two things:

1. You gain a good overview of how all workshop participants rate each assumption based on the dominant colors in a row.
2. Everyone can directly recognize assumptions that do not have sticky notes in the green area.

At this point, the question is not whether an assumption is correct. The sheer quantity of sticky notes says nothing about this correctness; a single sticky note described with good sources in the green area is enough to consider the assumption as knowledge. But this approach will help you identify the assumptions where you must determine *whether* they are correct as early as possible.

After a joint review of the cards in the second part of the workshop, a discussion follows. We go through each assumption and decide whether the assumption is well enough supported to treat as knowledge or not.

Any assumptions that can't be treated as knowledge after the discussion are then challenged in the context of use analysis (e.g., through user interviews). This activity is performed until you know that the assumption is correct, or it turns out that it is simply wrong. Thus, this assumptions workshop defines the questions you need to address at the beginning of the data-driven UX design (3DUX) process and determines which methods you choose. One tip: The more critical an assumption is to the success of your product, the earlier you should test it and the more effort you should invest in testing it.

However, the matrix you develop in this workshop will stay with you for the rest of the project. Whenever you discuss new features or receive further feedback, you can incorporate them into the matrix. For example, in the case of a new feature, you might consider whether you have enough data to justify introducing or rejecting the feature. Regarding recent user feedback received, you could check whether this data confirms or rejects an assumption. In the best case, the matrix accompanies you through the entire project.

6.1.3 Why Do I Need Data for This?

At this point, let's briefly revisit why the distinction between assumptions and knowledge should be made based on figures, data, and facts. From our point of view, we need data for several reasons, such as the following:

1. If you decide based on objective criteria based on data, the decision is easy.
2. Decisions based on data and objective criteria are easier to explain and defend to third parties.
3. Correct data makes you more resistant to biased perceptions of individual cases.

These three points may seem abstract at first, but in our experience, we've seen how they are omnipresent in practice and lead to significant or even insurmountable hurdles. Therefore, let's look more closely at each point next.

Point 1: Easier Decisions through Data

In our experience, few products are developed by a single person. In most cases, a whole team of people work together on the development as well as many other stakeholders—that is, people who are interested in the outcome of the development and would like to influence it in their own ways.

Since every stakeholder in development brings a different perspective and a different gut feeling, is sometimes intensively and sometimes less intensively involved, and in some cases still pursues individual goals, we often find our discussions become long and sprawling. This is especially true in project teams where two people find "their" design or "their" approach better without having a real reason. But this problem can also be true outside of project teams when, for example, the company's CEO says, "If anyone here knows the users, it's probably me!" Such discussions are usually counterproductive, and often product development comes to a standstill as a result.

However, suppose you have agreed, within the team, on several objective decision criteria that can be assessed against data. Then, you can clarify whether approach 1 is better than approach 2 and save yourself the lengthy, sometimes defensive, discussions.

We can illustrate this effect particularly well with a long-standing customer of ours. The team is responsible for the logon processes of a globally operating company. At one point in the development, various logon processes from different services were to be merged. The team had been discussing the best way to implement this feature for weeks. Half the team believed making the merging visible to users was imperative, so there would be no surprises. The other half of the team felt strongly that this was unnecessary because users would know the services were from the same provider. Then, in the workshop, we discovered that neither side had any data to support their theses. Once this lack of evidence was clear, the discussion ended immediately, and we conducted a user test to settle the issue.

Point 2: Easier Decisions to Explain/Defend through Data

If you have made your decisions with objective criteria based on data, as described in point 1, you can much more easily vouch for their correctness. You probably won't approach a CEO with a gut feeling—but rather with a well-founded data set. On the other hand, the CEO will be much easier to convince by your data-based argumentation (unless all we're doing is playing power games—then only a new CEO will help).

An example of the effectiveness of this tactic was demonstrated in a workshop with a medical device manufacturer. The issue was further developing an existing product that included a sealable tube. The customer support manager wanted the lid to open more easily in the next version of the product. The development team and the development manager saw no value in this change. The customer support manager then opened his ticket system and searched for tickets related to the tube. Within a short period, he produced over 20 tickets from the past year expressing issues with opening the tube. In some cases, these problems resulted in the spillage of potentially infectious fluids. As a result, it was recorded without comment that the force to open the tube had to be adjusted.

Point 3: Against the Biased Perceptions of Individual Cases

For this point, we must once again briefly digress into the world of psychology:

1. Your brain is obsessed with emotions.
2. Your brain attaches great importance to the order in which information is encountered.

Regarding the first sentence: We can more clearly remember things that were particularly positive or negative for us than things that only triggered weak emotions (see Tambnini et al. 2016). To put it differently: We remember extreme events much better than the neutral events in between. Our brain thus evaluates emotionally charged information as more valuable. In this way, the brain wants to help us experience positive emotions repeatedly and avoid negative emotions as much as possible.

Applied to our product development scenarios, you're much more likely to remember "extreme" customer feedback. Suppose you work with customers daily who always give you little or indirect feedback, and suddenly, you experience one customer who gets extremely upset about your product. After a few weeks, you'll probably remember this one customer in particular. Now, imagine that customers get upset with you about the product once or twice a month. At some point, as you rummage through your memory for customer feedback, you'll remember the many stories where customers were very disappointed with the product.

The same pattern works with enthusiasm. However, your perception can also be deceptive: You fade out the neutral opinions that may also exist. As a result, you may develop for a much smaller customer group that has just been mainly remembered—and neglect the "silent majority."

Regarding the second sentence, our memory falls for emotions and is strongly influenced by the order in which information is learned. The *primacy-recency effect* states that people remember information better when placed at the beginning or the end (e.g., Ebbinghaus 1913). For example, if you receive a 2-page letter, read it, and are then asked to reproduce the contents freely, you'll probably remember things more easily if the information were placed at the beginning or end of the letter.

If we apply this concept to the product development context, you'll likely remember more information from the initial and final user feedback sessions. Additionally, you may neglect the (larger) middle part or take mainly the extreme customer feedback only.

As you can see, even though our gut feeling is valuable and our memory helpful in many places, there is no guarantee that our memories are correct. But if you collect objective data, for example, during customer feedback sessions, you can also use this otherwise neglected knowledge.

We regularly experience an example of this effect in practice in usability tests where our customers are watching. Although several tests are carried out and the results are merged, we regularly experience customers weighing opinions heavily based on the first and last tests. A helpful counterpoint is to show how often problems actually occurred based on protocols to avoid bias.

Mindset: You're a Scientist

A vital mindset in usability and UX design is to see yourself as a researcher or scientist. You have assumptions and unanswered questions, and your task is to prove or disprove these assumptions and questions based on facts. This scientific approach also helps you develop and maintain a growth mindset. If you defend "your" solution, you can either win or lose. If you test assumptions and generate knowledge, you are primarily learning and growing.

6.1.4 Examples of Mostly-False-and-Seldom-True Assumptions

Around the topic of assumptions versus knowledge, let's explore a few typical assumptions we've repeatedly encountered in our daily work. The assumptions we'll now name and briefly explain are sometimes correct, but in most cases known to us, they are entirely wrong. It would be nice if, after this section, you could find the courage to critically examine yourself and your company to see if these are perhaps assumptions that you also make without checking them.

Our top-ten typical mostly-false-and-seldom-true assumptions include the following:

- The user speaks my technical language.
- We know our users because we've been developing their products for years.
- New technology alone is an excitement attribute.
- Users read the instructions for use.
- We work with experts, and usability and UX design is unimportant to them.
- Users are willing to learn about our product.
- Working software doesn't have to be beautiful; it must be functional.
- The technical requirements are much more complex and essential for product success.
- If my colleague can use it, so can everyone else.
- Usability testing will only reveal minor issues.

The User Speaks My Technical Language

Do you know the feeling? You're new in a company and need a few days, weeks, or months to understand what it means when the "CTO wants to have lunch ASAP with the COO about the project from ZDI so that everything can be BANFed in time for the deadline according to the roadmap."

We often feel like this, although not always, when we work with a new customer for the first time. Every company has its abbreviations, terms, and areas. All employees who have been with the company for at least a few months know and speak this language fluently. Often, this internal communication then spreads to the customer approach. Products are described using internal terms, the website asks the user to select the appropriate company department in the main menu, and suddenly, customers must understand you and your company instead of being understood by you.

This problem occurs even more strongly when the target group has an entirely different background than the employees of the developing company. Suppose young software developers develop a product for young software developers. In that case, the developers' understanding of the users is significantly closer than if young software developers develop a product for older software developers. However, if young software developers develop a product for older women (over age 65), their understanding will likely deviate quite far from each other. Error messages that are completely clear to software developers, such as a 404 error on a web server, usually trigger complete perplexity in the user group.

In addition, most products developed with usability and UX in mind are probably sold nationally or internationally, which also results in different conceptual worlds between developer and user and between user groups. Restricting ourselves to Germany, for example, when you order *Pfannkuchen* in Berlin, you get the pastry shown in Figure 6.3, on the left. The exact same request in Hesse gets you the pastry shown on the right.

Notice from these simple examples that the assumption "The user speaks my technical language" is often not true. But to know this for sure, you must test the product with the intended users. There is no way around this requirement.

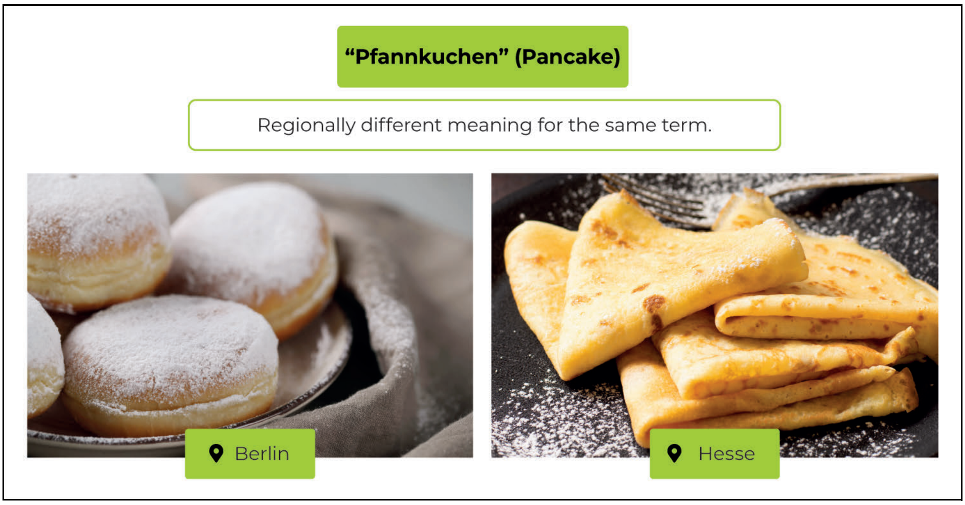


Figure 6.3 An example of the regionally different meaning of the same term: "Pfannkuchen" (pancakes) from Berlin (left) and from Hesse (right).

We Know Our Users Because We've Been Developing Their Products for Years

Developing products for users is not the same as knowing a product's users. Especially with companies that have been on the market for a long time, we hear this assumption quite often. However, in our kick-off workshop, one standard question we ask is "Who are the users of your product, and what tasks do they perform with your product?" Depending on the product, we also add a second question: "What other products are used together with your product?"

Can you answer these questions quickly and clearly? Would these answers be congruent with the responses of other members of your team? Would they be congruent with responses from your boss, the marketing department, or the development team?

If you're like most of our customers, the honest answer to all three questions is "no." This negative answer is not due to a lack of knowledge in the company per se but mainly because the knowledge is implicit and preserved by many different minds. Thus, no database contains any knowledge about the users, and everyone knows only one facet of the company's existing knowledge about users. A helpful activity is to bring new data into the process to speed up decision-making rather than laboriously gathering the data internally. This advice is especially the case since new data—perhaps even generated externally—is often judged as more valid than recollections from memory.

New Technology Alone Is an Excitement Attribute

Most of our customers have a technical background, and maybe you, too, are carried away by technology. Nothing wrong with that: We all have tech gadgets that we've anticipated or are still anticipating. For example, we are waiting for the launch of the *Virtuix Omni One*, a virtual reality gaming treadmill for the home.

In most cases, however, we see customers buying a product because of what it enables them to do, not primarily because of the technology built into it. Thus, before you elevate your product to a new technical base and are disappointed because customer feedback is cautious, ask yourself, "What are the benefits and drawbacks to our customers that we are now using a new technical base?"

If you can't cite customer benefits yourself, perhaps because you haven't passed on reduced maintenance/servicing costs, for example, or made the user interface (UI) more user-centric, you may need to understand your customers better. New technology is not an excitement attribute; it's what the technology enables your customer to do that creates delight.

Users Read the Instructions for Use

Hand on heart: When was the last time you read the user manual for any technical product? (We mean more than skimming the quick start guide.) Or when did you last sit down, read, and work through an app's tutorial step by step? For us, it's been a long time. The only user guides we read are the ones we revise and test for our customers, even when we've eagerly awaited a product.

We regularly conduct studies in which the comprehensibility of the instructions for use must also be tested (as is the case, for example, with medical devices). Even in these studies, when we explicitly ask the participants to answer questions based on the instructions for use, some participants admit to us that they would not read the instructions at home either.

For usability and UX design, you must assume that no one reads the instructions for your products! If, in reality, parts of your user group do read the instructions, only better results can arise than if you assume that everyone reads when really only some do.

When users reach for a user manual, they usually do so because they haven't understood something. Typically, they try some tasks out first and only look up the directions (sometimes angrily) when something doesn't work as desired.

Thus, if you assume during product development that the user manual is necessary to use your product successfully, you'll be met with little enthusiasm from your customers in most situations. Likewise, you can't use the user manual as a panacea for poor UI designs without alienating the users. So, when asked, "Do you think anyone understands this?" the answer should never be, "I don't know. Never mind, we'll explain it in the user guide."

We Work with Experts, and Usability and UX Design Is Unimportant to Them

Michaela here—this is one of my favorite assumptions! This assumption is plausible, and many experts diligently support it in conversation: "No, that's not important. I have 20 years of experience here, and I am trained regularly. It doesn't have to be pretty; it must work!"

At first glance, this assumption may seem entirely justified, and you may even encounter resistance from the experts against making an existing, complex system more accessible. Can you imagine why? Because of fear! A quiet voice in the back of their head is shouting, "If the product were easy to use, then they wouldn't even need me as an expert!"

In this context, too, the key to success is to remove this fear and focus on the added value for the users. Why not ask your experts how much time they put into operating the product? Or how long a new employee takes to make a difference? Our company specializes in complex usability and UX issues: We are in the medical market, in the control of large live broadcasts, in emergency landings of airplanes, and in process controls where many people and many millions of euros are involved. Almost all of our users are experts.

In fact, many of our customers have become market leaders *because* they have given the experts back the time to do what they are good at: their jobs, not operating a product!

Users Are Willing to Learn Our Product

You're partly right if you think our top ten list is wrong by including this assumption. This assumption is not always wrong. In some contexts, it is natural for training or familiarization to occur. For example, pilots train intensively with the emergency landing system we've mentioned in earlier chapters.

Nevertheless, we've observed in recent years that the willingness to invest time has noticeably decreased due to an increasing work density. If, for example, three training days used to be sufficient, now, it might be 3 hours, although 30 minutes is preferable, preferably via video. Both companies and participants share these expectations about training. In your private life, when was the last time you had to complete a training course lasting several days—or just several hours—before you could use a product? And when was the last time you were happy about it?

Our recommendation is to assume that users are looking forward to your product's possibilities but not looking forward to investing any time in learning how it works. Thus, if alternatives are available, users will quickly jump ship and try another product when they realize they must learn the ropes. Perhaps you have had this experience with an app in the app store. If the app is too complicated, you install an alternative instead of learning how to use it.

Working Software Doesn't Have to Be Beautiful; It Must Be Functional

Again, this assumption is not fundamentally wrong: Many users are willing to accept compromises in the visual appearance of applications in a work context. But we promise: The topic of visual design and usability comes up in every conversation—actively encouraged by users. Many applications are then judged to be old-fashioned or simply no longer modern. And most importantly, users often wonder, "Why does everything look great on my smartphone while I sit all day on ugly software that also costs a lot more than these apps?" Nevertheless, the applications are used, of course. It is work, after all.

In the meantime, however, enough studies and literature has proven that software that has not been developed in a user-centered way increases stress on employees in both the short and long term and is even rated as a risk factor for mental stress (Hurtienne & Prümper, 2003). Modern work equipment is also increasingly important in attracting young employees (IDG Research Service, 2018). Especially for Generation Y, poor equipment is a ready reason for quitting (Mindspace 2018).

Technical Requirements Are Much More Complex and Essential for Product Success

Whether and to what extent this assumption is valid depends primarily on your product, system, or service. Of course, some solutions have technical requirements that are very complex. Then, the ability of your solution to enable certain activities represents the basis of success.

Technical features are usually enough if you're entering an entirely new field with no competition yet. At present, no one expects to be able to fly a rocket to Mars without training and education. However, if your product is complex but not unique (anymore)—for example, an industrial plant control, a process control, or an Internet of Things (IoT) application—then your users will compare solutions.

We regularly experience that, in any specific market, more or less all manufacturers in that market will largely cover the same technical features. The differences often lie in the details, which are no longer critical for most customers—even if these differences may have been significant before. If the technical differences are minor and are the only differences, much explaining and convincing will be required to get users excited about your solution.

For example, a high-quality visual product design can help exploit the what's called the *halo effect*.

In psychology, the halo effect occurs if a person infers another characteristic of another person based on one characteristic ("Halo Effect," American Psychological Association: Dictionary of Psychology, n.d.). For example, many assume that a person in expensive brand-name clothing is professionally successful. Unfortunately, this characterization also works the other way around: Many would attribute the characteristic "lazy" to a severely overweight person. The halo effect occurs, especially when decisions must be made quickly.

Why shouldn't this impact be similar for products? If you must decide whether to take a closer look at a product, its visual appearance plays a significant role because it can be grasped quickly. A product that looks high quality is then attributed to additional premium features. These features, however, cannot be captured at a glance. A higher probability that you'll be given more time in the selection process exists for products that look better than others.

If My Colleague Can Use It, So Can Everyone Else

Instead of colleague, you can say spouse, mother, or any other conceivable person who has no relation to the product and the task—but to you. If your colleague belongs to the product's target group, then the assumption can be correct in principle. Especially for products with very broad user groups (e.g., a communication app), feedback can also be helpful. However, as soon as a specific task is behind a use that is only performed by a particular user group (and not by your colleague or your mother), you can no longer do much with the result.

In addition, other significant effects must be considered: Remember our earlier point about the user speaking "my" technical language. A colleague also speaks the company's technical language and may not have any problems in places where other users get stuck.

The problem of the *friendly user* occurs again and again in everyday practice. If you ask someone close to you, that person doesn't want to hurt you and wants you to succeed above all else. Or maybe you have a straightforward conversational culture with your friends and regularly receive a full broadside of problems, mistakes, and failures. You see where we're going with this. Your product is often rated differently by friendly users than by people who don't know you. Since you probably don't personally ship yourself with your product to be close to every user, this feedback doesn't help you much.

Therefore, our advice is to critically examine for yourself and your company whether you're talking to the right people regarding realistic feedback on your product. Select the people giving you feedback solely based on whether they are among the users, not primarily on how easy it is to get their feedback.

Usability Testing Will Only Reveal Minor Issues

Teams and companies that have never conducted user testing often assume that user tests will only uncover minor issues: "The overall concept is in place, the features fit, and at most, a few buttons need to be moved from left to right or a little tweaking to the look and feel." Relying on this assumption can work but rarely does in practice.

As you'll see in later chapters, a user test is for learning, not for "passing." If you've never gathered real feedback on the product, then you've been—to put it bluntly—guessing all along. How likely are you to guess correctly in such a complex field?

If you have already conducted a lot of user tests, perhaps with a predecessor product in the past, and now it's time for a remake, then you can use this strategy. However, notice the big "if" here.

In practice, we repeatedly encounter that people don't believe us. No matter how well we argue—they somehow assume that their product is the exception. In practice, this false assumption leads to us repeatedly seeing panic-stricken postponed release dates and hundreds of developer years wasted.

For all parents among our readers, let's make a slightly different comparison at this point. Almost everyone without children feels that having children can't be that exhausting and that you'll manage. You can't imagine that it's actually as incredibly exhausting (and incredibly beautiful) as all the parents have told you. But as soon as you have your first child, you quickly realize all those parents were right.

The same is true with user tests. As long as you haven't experienced a user test on your product, you may think testing is unimportant. After you conduct your first test, however, you'll ask yourself how you could ever work any other way and why you waited so long.

Therefore, we strongly recommend taking feedback seriously and scheduling initial user feedback as early as possible. The earlier you test your product, system, or service, the easier making changes will be, thus preventing frustration, budget increases, and postponed deadlines.

Thus concludes our top ten list of mostly-false-and-seldom-true assumptions. And? Did you recognize yourself in one or another? Don't worry! Identifying your assumptions is the first step to developing an even better product. But we have one more assumption for you—one so important in practice that we've dedicated an entire section to it. Let's jump right in.

6.2 Users Don't Know What They Want!

This assumption is—in one way or another—repeatedly advocated by experts of all stripes. Even a very well-known quote by Henry Ford revolves around precisely this assumption. He is credited with saying, "*If I had asked people what they wanted, they would have said faster horses.*" This favorite phrase of design experts and developers is used to justify developing what they personally find valuable and beneficial. In this section, we would like to present our view on this topic. We are eager to hear what you think about it.

6.2.1 Task and Goal Expertise versus Design Expertise

We can completely agree with the quote from Henry Ford. This precise scenario has played out repeatedly in our conversations with users. When asked, "How should we design the product in the future?" the user responds, "The same as it is now, only with fewer bugs" and maybe "It would be nice in blue, too."

This type of conversation leads to multiple outcomes, often in precisely this order or a very similar order:

1. Users feel helpless because they cannot say what the product should be like.
2. The developer feels, "Oh great, I didn't have to ask."
3. The development team does what it sees fit and focuses on technical features and bugs.
4. In the end, the users are still not enthusiastic.
5. The development team decides not to involve users in the development in the future because doing so only costs time for no impact.
6. A design is chosen that the development team is convinced of in the absence of other design cues.
7. In the end, the users are still not enthusiastic.

This downward spiral often results from confusing *task* and *goal expertise* with *design expertise*.

Let's clarify the difference with an extreme but hopefully catchy example: Imagine you're one of the world's leading biologists and know more about plant growth than anyone else. Now, a far-reaching space project is set to explore how well plant growth works in space. In preparation for the space mission, the development team for the rocket is now coming to you and asking you, as an expert, how it should build the rocket. Could you tell the development team how to construct the rocket? This example is extreme, but why should it differ for an app, a process control, or a medical device?

As a biologist, you're being sent into space because you're exceptionally knowledgeable about plant growth, not because you're particularly good at building rockets. The problem that users do not contribute to the design is, therefore, not a problem at all because the involvement of the users is not valuable in that area. Rather, the way it is done ("How am I going to build this?") is not sufficiently meaningful on its own.

Let's take another look at the different types of expertise. We call the expertise of those content experts *task expertise* and *goal expertise*.

Task Expertise

Task expertise is the users' knowledge about the task they want to solve. A task can be predetermined (e.g., at work) or self-selected (e.g., in leisure time). So, in our example, the biologist knows a lot about plants and what these plants need for optimal growth.

Goal Expertise

To explain *goal expertise*, we must backtrack a little: Needs generally express what a person craves or wants to achieve. Looking at typical needs pyramids or hierarchies of needs, they usually start pretty broadly at the bottom with fundamental, physiological needs (e.g., eating or drinking) and become less and less tangible and abstract at the top (e.g., self-actualization).

These ideas first came from Abraham Maslow (1954), whose the hierarchy of needs, shown in Figure 6.4, has been taken up and explored over the years. Sheldon and colleagues (2001) have more recently identified significantly more needs relevant to the use of technology.

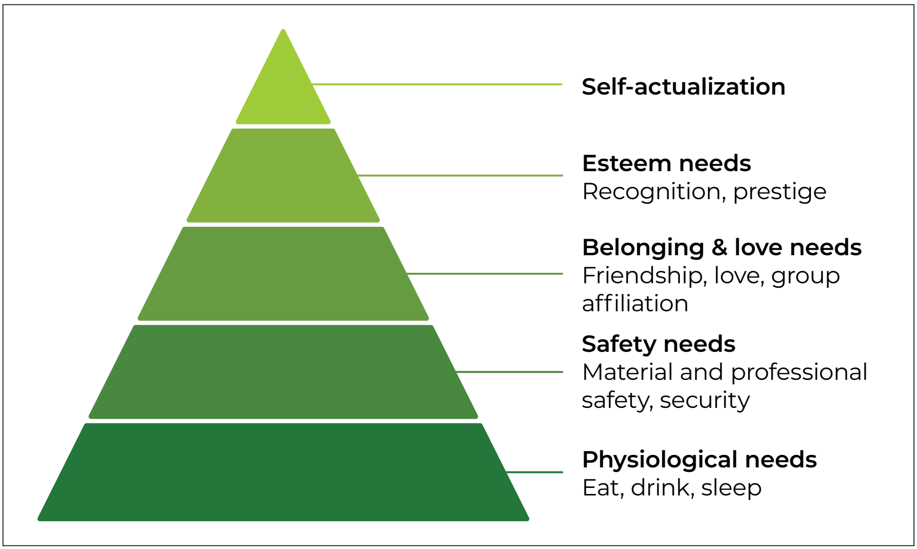

Self-actualization
Esteem needs
Recognition, prestige
Belonging & love needs
Friendship, love, group affiliation
Safety needs
Material and professional safety, security
Physiological needs
Eat, drink, sleep

Figure 6.4 The hierarchy of needs with its five levels. (Source: Based on Maslow, 1954)

The field of needs thus represents a general framework that roughly describes what people can strive for. These needs can be conscious or unconscious and impact a person's goals. These goals then break down quite specifically for an individual person, including how these needs will be realized for that person.

The goals can therefore be concrete compared to needs. For example, a biologist might set a goal that a specific plant will bear fruit within 6 months. The need behind this goal could be a quest for social recognition ("Ah, he can make this plant grow in space!").

However, goals can be directed not only at external events or observables but also at one's perception and feelings. For example, the biologist might aim to shield himself during work and work in complete silence.

One approach that brings needs and goals together is Christensen et al.'s (2017) *jobs theory*, which defines "jobs" at three levels:

- **Functional level**
  The functional level contains the observable goal (i.e., the goal that third parties could also see, hear, or experience). If we stick to our chosen example, we can classify the goal of making a specific plant bear fruit within the next 6 months as a functional goal.
- **Social level**
  The social level revolves around how a person wants to be perceived by others. These goals can be partly inferred from the outside and partly not. If we think of our biologist on a space mission, then one goal on the social level could be the desire to be perceived as competent by the other astronauts and not as a burden.
- **Emotional level**
  Finally, the emotional level is about how a person wants to feel during use. This level can't be accessed through external observation and must be inquired about by asking the person. The biologist wanting to pursue a fulfilling activity and having fun at work can be classified as emotional goals. In addition, the biologist could want to feel safe in space.

As promised, we now return from the excursion into theory back to practice. How does all this discussion help you in designing inspiring products?

In straightforward terms, as shown in Figure 6.5, the functional level tells you what features your product must bring. In other words, it answers the question, "What must it be able to do?" The response must be based on the *hard facts*, for example, that a farming station exists in the space station—there is no way around that requirement.

The other two levels tell you *how* your product must implement the functional-level features for the user to be successful. How do you achieve with your farming station design that the biologist is perceived as competent? How can you contribute to him having fun at work? These are precisely the kinds of questions addressed at these levels, and in the vast majority of cases, answering these questions results in new features at the functional level.

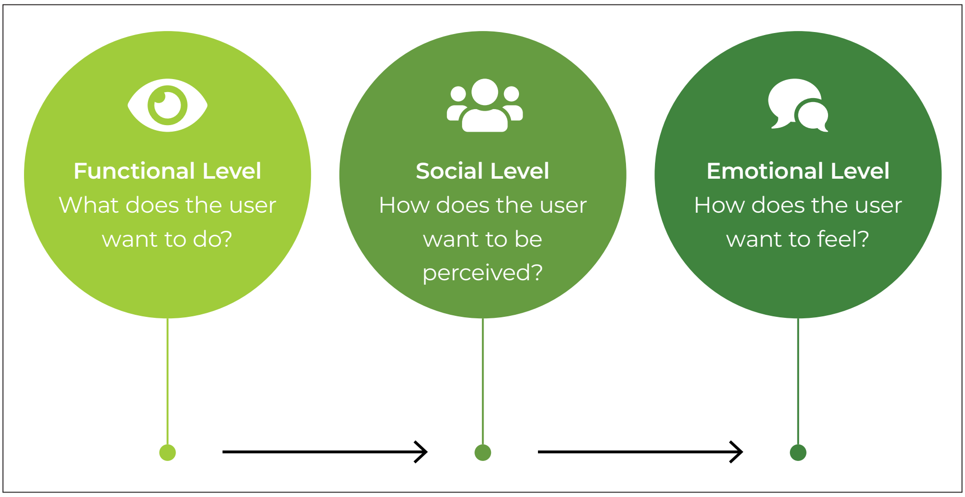

Functional Level
What does the user want to do?
Social Level
How does the user want to be perceived?
Emotional Level
How does the user want to feel?

Figure 6.5 Subdivision of goals according to the three levels: functional, social, and emotional.

Summary of Task Expertise and Goal Expertise

Both task expertise and goal expertise make up your users' overall expertise. To develop a product that inspires your users, you must consider all three levels of goals. Direct contact—through observation and questioning—is essential to tap into these goals.

Note that some goals are unconscious and, therefore, difficult to verbalize. The question "How should we design the product in the future?" is helpful in very few design projects. Transferring needs, tasks, and goals to design requires different expertise: design expertise. However, the assessment of the design by users is a reliable source of information. If you hear "This doesn't work for me," then you can be relatively sure that it won't work.

Design Expertise

Under design expertise, we summarize the skills you need to develop suitable solutions from user requirements, which may involve technical or design solutions, depending on the requirements.

Often, a whole team works together during development. You may have a software architect, someone for the backend, someone for the frontend, and someone else who does IT security. Most technicians know that different competencies are needed, and no IT security person would think of asking the frontend person about the best encryption algorithms.

6.2.2 Bringing Together the Expertise

Now, we come to the core of this chapter: Many in development are aware that users must be involved in development. But some wonder what exactly they can contribute.

Generally speaking, the people who will use your product are the experts on their tasks and goals. In contrast, you and the other designers and developers are experts in solution design. Thus, when Henry Ford states that people would have wanted faster horses, and that information doesn't do him any good, it's because he asked the wrong questions. The need—to get from one place to another more quickly—he could have (and did) easily elicit (the expertise of the user). Observation and questioning methods are particularly suitable for identifying user tasks and goals. When observing, you focus on the questions “What do the users do?” and “How do the users do it?”

When questioning, you mainly talk about:

- How the decision for or against a specific solution was made
- Which alternatives were still available and why these are not eligible
- What has already been tried but found not good enough
- Why a particular solution is perceived as good or bad

In other words, you focus on areas in which your users are experts. For more information on how to proceed in detail, see Chapter 9, Section 9.3.

The (technical) solutions, on the other hand, must not come from users but from designers. If a miracle pill could have been developed to speed up horses or to breed a much faster horse, then they could have gone back to the users with these ideas (and the concept of the automobile) and received feedback on whether these ideas solve the problem, or which is the best solution.

Looking at our examples, it's clear that moveHUB should learn more about the goals of its users, including, for example:

- Where do they want to travel (e.g., city to city, city to country, country to city, country to country)?
- What do they value in travel?
- What travel alternatives are they using right now?
- What makes current offerings good or bad from the users' perspective?

All these questions are part of the expertise of the users. The answers to these questions—obtained through observation and questioning—then provide information on the three levels (functional, social, and emotional). Results on the functional level could include the following requirements:

- An overview of all possible travel offers must be displayed.
- The cost of each alternative must be visible.
- The travel duration for each travel alternative must be clear.

On the social level, findings may inspire the following requirements:

- Users always want to know what to do next (don't want to ask for help → display of competence).
- If someone chooses the cheapest alternative, this choice should not be directly apparent to others (don't want to look like they are concerned about money → display of social status).

On the emotional level, findings may inspire the following requirements:

- Users do not want to rush from one means of transport to another while stressed (target state: relaxation/rest versus stress).
- Depending on the current situation, users may wish to travel alone or with others (need for closeness or social exchange).

Based on these findings, various implementations can then be sought, which should be proposed by the design team. For example, a filter option could be created to switch between individual travel and group travel options based on the desire sometimes to travel solo or with company. However, the design team must decide how to implement these options and get user feedback.

BusinessBooster, our combined customer relationship management (CRM) and invoicing software, already has customers and can therefore get an initial picture of usage via usage statistics for various functions. During the first user contacts, BusinessBooster should focus on getting to know the users' tasks in detail, thus immersing itself in the world of user needs and goals. Even BusinessBooster cannot expect users to design a perfect software interface—but an understanding whether the functions offered match the user's expectations can be easily obtained.

Currently, the Healthyfy physician and patient portal is built heavily on assumptions. These assumptions should be reviewed with users. For example, Healthyfy could ask patients and physicians whether such a solution is conceivable. We're not yet talking about details, such as the placement and color of a button in the chat function; we're more focused fundamentally about whether people would even use Healthyfy with the concept envisioned.

6.2.3 Creating Outstanding Solutions with Users

Thus, both sides are needed for successful product design: on the one hand, the users with their extensive and detailed knowledge of the goal and ability to distinguish good and bad solutions—in other words, task expertise and goal expertise. On the other hand, you need design expertise that turns requirements into solutions and can adapt existing solutions based on user feedback.

This realization leads, in our opinion, to another logical conclusion: You must have at least two types of expertise in usability and UX design as well!

On the one hand, you need expertise to determine the users' tasks and goals and whether a solution is good or bad. On the other hand, you need expertise to design suitable solutions. In our company, we explicitly divide this expertise into two different areas:

- User research
- Usability and UX design (i.e., the creation of solutions that have a high level of usability and a good UX)

Usually, you won't find these competencies in the same people. In *user research*, you often see people who have a human-related background (e.g., psychologists, sociologists, anthropologists, or even marketing and social research specialists). In *usability and UX design*, you often meet designers, engineers, or software developers. We'll go into more detail about these fields and the mindsets that help work in these areas later in this chapter. In general, think of user researchers as translators of user language into requirements and usability and UX designers as translators of requirements into solutions.

6.3 Key Skills for User Researchers: Observe, Ask Questions, and Listen

In the previous section, we discussed two areas of expertise needed to create a product, system, or service with outstanding usability and UX. In this section, we would like to take a deeper look at the first one: user research.

If you want to work as a *user researcher*, you should like dealing with people—because interacting with people will be your main task. Empathy can hardly be overestimated either, to identify users' emotions, thoughts, motives, and personality traits more easily. So, for user researchers, there are essential characteristics that facilitate the required activity—generating knowledge about users.

Regardless of the individual characteristics, however, three outstandingly critical skills exist. We call them the *mega-skills* of the user researcher:

- Observing
- Asking questions
- Listening

To the chagrin of all human experts and user researchers, these skills sound pretty easy to learn at first glance. Probably most people would even claim they have mastered all three skills without problems. After all, we do everything in everyday life: observe, ask questions, and listen. Perhaps you, too, have just caught yourself thinking, "Well, it can't be that hard."

Unfortunately, these misconceptions are common. Unlike designers, who are assumed to have innate talent, user researchers are often disregarded for their skills until you first see a trained user researcher at work. So, what distinguishes the trained user researcher from a "normal" person? It is the quality in which they observe, ask questions, and listen. Let's start with observing.

6.3.1 Observing

Please briefly consider the different areas of expertise distinguishing users from developers or designers (Section 6.2.1). On the one hand, we had the *task* and *goal expertise* with the user, and on the other hand, the *design expertise* with the product development team. In the goal expertise area, we introduced you to the functional, social, and emotional levels. We also discussed that not everything can be observed, especially on the emotional level.

In our understanding, the user's world is divided into an external world (what the user does) and an inner world (what the user thinks and feels). To capture the user's external world, the tool available to you is observation.

The Different Types of Observation

You can observe users directly or indirectly. With *direct observation*, you or another observer are personally on-site and watch. *Indirect observation* is conducted with a camera, you evaluate the recording either live or later.

Furthermore, observations can happen overtly or covertly. In an *open observation*, the observed persons know they are being observed. In a *covert observation*, the participants do not know.

The different types of observation have advantages and disadvantages: In-person direct observation is more flexible because you can follow a person around if they do something unexpected or unplanned. Therefore, you don't have to think about exactly where to place cameras so that you can observe all the parts of the user's task you want. On the other hand, indirect observation with cameras is less disruptive than face-to-face.

With open observation, you have the advantage that no legal or ethical issues arise—after all, users know they are being observed and can consent. This consent is not possible with covert observation. On the other hand, with an open observation, users may change their behavior compared to regular usage. For example, a user might try to do a task exceptionally well, which might not be the case without observation.

In contrast, in the case of covert observation, users behave the same way as in the unobserved case because they are unaware of the observation, which raises legal and ethical questions.

Legal and Ethical Issues in Covert Observation

A word of caution: Especially if you take footage of people without their knowledge and permission, you put yourself in a difficult position legally. In public environments, covert observation may be possible if the focus is not on a single person but on a place with many people. However, you should always research legal aspects before deciding on covert observation. Invading a person's private space without their knowledge is always wrong.

In addition to the legal aspects, there is the ethical question of whether collecting data on individuals who have not consented to this data collection is okay. Many journals now no longer accept studies based on unapproved data.

Of course, you can also combine both types of observation and observe open/direct, open/indirect, covert/direct, and covert/indirect. However, you must not influence the situation with your presence if possible.

Objective Documentation versus Direct Interpretation: Why User Research Needs Experience (Part 1)

No matter how you perform the observation, the goal always remains the same: You want to use observation to obtain the most objective data about what a user is doing and how. "Objective" means that different people who see the same scene would come to the same description (see Chapter 5, Section 5.6.2). So, the first step is not interpreting what you see, just documenting what happens.

This objective documentation is precisely where the quality of a trained user researcher becomes apparent, whereas the objectivity of inexperienced observers is often lacking. Inexperienced observers often place what they see directly in their sphere of experience and interpret it. As a result, objective data is not collected and evaluated, but conclusions are drawn directly and documented instead (see Chapter 5, Section 5.6.3 and Section 5.6.4).

For example, the creators of moveHUB, our travel app, might have considered, as a basic assumption for their app, that a delay in public transportation annoys passengers. This annoyance triggers users to take out their smartphones and look for alternative transportation with the moveHUB app.

To test this assumption, the founders observe users at a train station. When a train is enormously delayed, the founders observe that all seven users waiting on the platform are obviously annoyed. The founders are happy to see their assumption confirmed and document the observed anger. Unfortunately, the early interpretation had the effect that the founders did not observe and document the subsequent events.

If they had done so, they would have documented that three of the seven people pulled out their smartphones and started a game to bridge the waiting time. Two people also took out their smartphones and ordered a cab. The remaining two people went to the train service counter to look for alternative travel routes.

This objective documentation (which urgently needs to be supplemented by further observations of delays but which we neglect for this example) allows an entirely different interpretation in retrospect: Although all persons are annoyed, not a single person uses their smartphone to search for alternative connections! If this picture is confirmed with similar observations in other stations, the evidence speaks against an app as a solution to the problem.

The objective documentation, separate from the interpretation, also allows moveHUB's founders to look at the data again later and, combined with new insights, also leads to a different interpretation. However, suppose the founders have only documented the interpretation ("All people get angry, and thus our assumption is confirmed"). In that case, the objective raw data is missing, and other interpretations become impossible (see Chapter 5, Section 5.6.2).

Outlining an Observation: Why User Research Needs Experience (Part 2)

In addition to objectivity during an observation, you can also recognize a good user researcher by their ability to break down an observation meaningfully. Depending on your observation goal, you can observe a person, the environment, or a process. The differences are quickly explained:

- When observing one *person*, you document only the actions of that one person and follow that one person.
- When observing the *environment*, you watch it change, including who or what comes and goes, but you stay put.
- When observing a *process*, you follow a task even if it spans multiple people and environments.

Let's look again at moveHUB, our travel app, to illustrate these differences. If moveHUB's founders want to learn more about their users (i.e., the person), they follow a user using public transportation. In this approach, they might observe which mode of transportation is used, what time this happens, how the selection takes place, etc.

If the founders are trying to understand the environment, they could place themselves in a train station and observe how many people are in the station, what these people are doing, how they get to the trains, etc.

However, if the observation focus is a process, the founders could, for example, follow an angry train passenger whose train has been canceled due to technical difficulties as they go to the service counter to complain. After he has handed in his ticket, the founders follow the complaint process further by observing what the employee at the service counter does with the complaint. Next, they might track the release of a compensation payment as well as the payout process itself back at accounting at the railway's HQ.

As you can see, "simple" observation can suddenly become complex and require planning. A good user researcher does not just observe randomly but selects suitable observation targets and methods for checking their assumptions so that the user research leads to the desired knowledge gain instead of just gathering "any" data.

In Summary: When Is an Observation "Good"?

You perform a good observation when the following conditions are met:

- You have little or no influence on the behavior of those present through your presence. In other words, you should try to be as inconspicuous as possible so that what you observe corresponds to what you could observe if you were not there.
- You document the objective facts and not your interpretation of the observation. This approach allows you to compare different observations with each other in the follow-up to the observation. It also enables you to identify patterns and similar behaviors, which only works error-free if you compare the behavior, not your interpretation.
- You have thought about and planned what you want to observe and why. In this context, you should be guided by the assumptions that you want to check.

As you can see, the methodologically clean approach to user research (called "research" for a reason) is critical to successfully gathering the knowledge you need to design for outstanding usability and UX. So, when you think about your approach to usability and UX design, we recommend considering user research as important as any other component.

Advantages and Limitations of Observations

In the external world of users (i.e., their observable behavior), observations are the most reliable data source. Unfortunately, asking people about their behavior without observing them only works to a limited extent. The reason for this limitation is that a large proportion human behavior is unconscious. This unconscious behavior is often highly trained and thus automated. Sometimes, therefore, users find it challenging to describe correctly what they are doing, even when asked. The description will then mainly include the parts that are not highly automated and about which the user must think consciously.

Observation is a fantastic method and belongs in every user researcher's toolbox. Unfortunately, the limitations of observation are as clear as the benefits. You can only observe what is visible. Therefore, an event must occur while you are observing. But what happens if what you want to observe is in the past, is not happening right now, is not allowed to be observed by you, or perhaps has not happened yet? Now, the second mega-skill of the user researcher comes into play: asking questions.

6.3.2 Asking Questions

Asking questions suffers from the same problems as observation. Probably you agree that anyone can ask questions. “It’s not that hard, right?”

And this prevalence is precisely the problem. Anyone can ask questions, but the quality of the questions varies greatly. The question in the previous paragraph is what’s called a *suggestive question* because it doesn’t ask for an opinion but instead puts your own answer into the question itself. As a result, depending on the other person’s personality, you may gain (false) agreement because the person just doesn’t want the potential drama from your reactance.

Reactance is the psychological term for

> *“a motivational state characterized by distress, anxiety, resistance, and the desire to restore that freedom. … when people feel coerced into a certain behavior, they will react against the coercion, often by demonstrating an increased preference for the behavior that is restrained, and may perform the behavior opposite to that desired.” (American Psychological Association, n.d., “Reactance”)*

In other words, if you ask suggestive questions and your counterpart feels that you want to hear a specific answer, he may have the same opinion by chance, agree with you to keep the peace, or get angry. Regardless of what occurs, good results from suggestive questioning are thus only obtained by chance.

Open Questions

To ask good questions, an essential task is to keep the goal of the questioning in mind. Are you still at the very beginning and rather explorative? If so, you might want to learn many new things and still have little data. In that case, the use of *WH questions* is essential.

WH questions—also known as *open-ended questions*—summarize the questions that freely ask for the opinion and assessment of the other person. Typical WH questions begin with the following words:

- What
- Who
- Whose
- Where
- When
- Which
- How often
- How much
- Why

The goal of WH questions is to ask questions in such a general and open way that your counterpart is encouraged to talk while remaining unaffected by your expectations.

Turning to moveHUB, an appropriate question might be, “How do you get to work in the morning?”

If you want to learn more about patient behavior for Healthyfy, you might ask, “What do you currently do when you wake up in the morning and realize you’re sick?”

Both questions are asked quite openly and do not yet have a direct point of contact with the solution. We don’t ask about public transportation or finding a doctor as a first step. For a successful product design, a crucial step is to understand whether the solution we would offer fits at all. However, we can only determine this fit if we know our users’ “natural” behavior. Habits are challenging to change, and the benefits of a change must be substantial for change to happen. Thus, a more successful approach to product development is to start with users’ habits rather than changing them.

In the course of questioning, the questions become more and more detailed; you “drill” into the topic. Good user researchers pick up on the answers and use the other person’s words for further questions. This approach increases the *rapport* (a warm, relaxed relationship of mutual understanding, acceptance, and sympathetic compatibility between or among individuals; American Psychological Association, n.d., “Rapport”) between the interviewer and interviewee. As a result, the interview is more pleasant and open, leading to better insights.

Furthermore, rapport is increased by additionally mirroring the behavior of the counterpart. For this purpose, body posture can be adapted to that of the other person during the conversation. For instance, if you’re talking to someone sitting reclined in an armchair, you’ll increase rapport if you adopt a similar posture. However, this advice only applies to positive mirroring. If your counterpart is depressed, hostile, or dismissive, you should not mirror this behavior but instead try to remain open and approachable anyway.

You may have noticed that we put the word “Why” at the end of the list—not because questions starting with “Why” are bad, but because of one thing you should note. You must ask a why question in such a way that the interviewee is not justifying the situation. If you think this might be the case, we recommend that you replace these questions with other phrases, such as the following:

- How did it come about that you...?
- How did you decide that...?
- For what reason did you...?

You can use these questions whenever you want to gather new information. Sometimes—admittedly relatively rarely in product design—you may just want to know which variant users prefer. Then, you can choose *alternative questions*. A typical alternative question might be “Do you prefer variant A or B?” We admit we’re WH question enthusiasts. When we use alternative questions as part of a face-to-face survey, we always follow up with a question like “For what reason...?”

A great strength of questioning compared to observations is that they allow you to understand the “why” of users—at least for conscious decisions. If you limit yourself to the “what” (e.g., choosing a preferred design), you lose this added value that questioning provides.

Closed Questions

Another type of question—to which, strictly speaking, the alternative questions also belong—is called a closed question. Closed questions have predefined answer categories. You can offer several custom alternatives as earlier, but you could also simply offer “yes” and “no.” Closed questions are quick and easy to answer and allow you to select and quantify. However, they do not bring you new answers.

Open questions are particularly appropriate for face-to-face interviews and group surveys, whereas closed questions are most useful for written surveys.

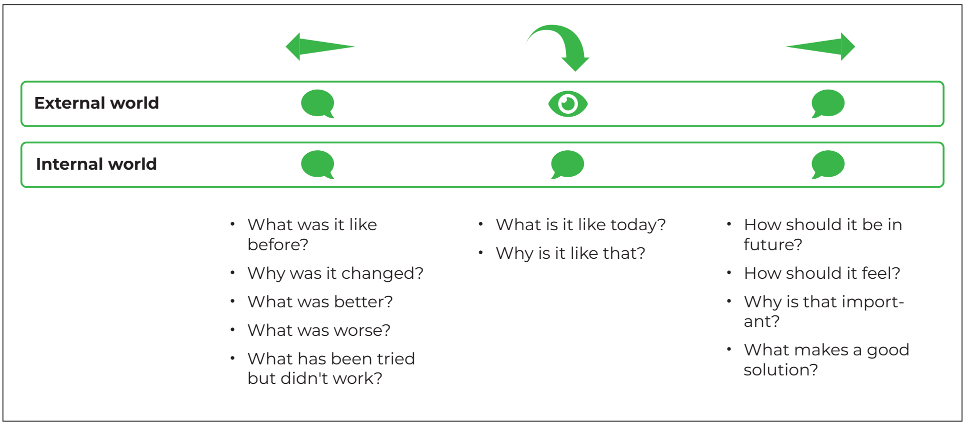
How to Sort the Questions Correctly

In addition to asking the right questions at the right time, the skill of asking questions also involves formulating and sorting the questions in chronological order. We recommend that you always ask questions from the general to the specific. So, using moveHUB as an example, first ask how someone gets to work before you talk about the advantages and disadvantages of bus transportation and then ask about a specific bus.

What Do You Ask About?

Another strength of questions is that you can also ask questions about things that have happened in the past, things that will happen in the future, and something that may not happen at all—in other words, they are purely hypothetical. This way, you expand the spectrum of user research.

Figure 6.6 shows which dimensions of questioning and observations can be served.



- What was it like before?
- Why was it changed?
- What was better?
- What was worse?
- What has been tried but didn't work?

- What is it like today?
- Why is it like that?

- How should it be in future?
- How should it feel?
- Why is that important?
- What makes a good solution?

Figure 6.6 Question structure according to the external and internal world and the past, present, and future. The graphic additionally shows what is observable and, thus, what possibilities questioning offers.

How Can You Recognize a Good Question?

As a rule of thumb, good user research is characterized by the user talking about 80% of the time. Don't worry; This won't feel strange to your counterpart. People love to talk about themselves. You can fill up an entire conversation with questions without having told a single thing about yourself, and your interviewee will feel like this was the best conversation they've had in a long time. This tactic is explicitly used by FBI hostage negotiators (see Voss & Raz, 2017). So, take heart!

Asking the right questions is a real challenge. If you would like to learn even more about questions, we have a whole list of further reading to recommend:

- The book *Change Your Questions, Change Your Life* by Marilee Adams (2022) deals with the topic of questions in a reasonably general way.
- If you're looking for good questions, we have had excellent experiences with resorting to the coaching field. Especially in systemic coaching, you work with questions to make the counterpart think. Of course, not all questions are suitable, but we are sure you'll find some inspiration. The book *Coaching Questions: A Coach's Guide to Powerful Asking Skills* by Tony Stoltzfus (2008), for example, contains over a thousand coaching questions, some suitable for identifying requirements for good solutions.
- For difficult conversations and challenging interlocutors, we recommend the book *The Moderators Survival Guide* by Donna Tedesco and Fiona Tranquada (2013).

To ensure that your results are comparable and that you can identify patterns in the follow-up, we strongly recommend creating a guideline for each survey that includes the core questions. We use *semi-structured interviews* in most cases. In those interviews, you use a guide containing questions that will be asked of each participant in any case. Additionally, other appropriate questions can be asked depending on the situation. Semi-structured interviews thus represent a good mixture of free conversation and systematic information gathering. In the best case, your counterpart will not notice that you're following a guideline, but it will feel like a normal conversation.

In Summary: Good Interviews

You'll do a good interview if you do the following:

- Use predominantly open questions, moving from the rough to the fine
- Record these questions in a guide to ensure comparability of the surveys
- Manage to create a natural conversational situation in which the user speaks about 80% of the time

The last point—the user talks about 80% of the time—also clarifies why listening is our last mega-skill. Listening is not as easy as you might think at first. Why? You can find out now.

6.3.3 Listening

When you're a user researcher, you have the challenge of getting people you don't know to tell you a lot of personal information quickly. You also want to get as much information as possible from the conversations. Therefore, you're documenting on one side and have a long list of questions to discuss with your counterpart on the other.

But now, imagine the following situation: You're invited to a conversation, enter the room, and sit down after some small talk. The interviewer has a piece of paper, a pad, and a pen and starts reading the first question. You begin to answer, but your counterpart doesn't look at you at all but is intently writing down everything you say. When you have finished your answer, your counterpart continues writing for a while, then looks at his notepad and asks the following question without looking up.

How do you feel? Do you think you would feel comfortable in this situation and would enjoy answering questions? Would you feel your counterpart is interested in your answers? Probably not. Here is where the mega-skill of listening comes into play. Of course, the person in our example is also listening. How else would he be able to take notes? But he does not give you the feeling that he is listening.

Active Listening

To give an interlocutor the feeling of really listening, active listening helps. Active listening is characterized by actively confirming what is being said, summarizing, asking questions, and being approachable.

Active confirmation primarily means verbal expressions such as "hm," "um-hmm," and "uh-huh" (in technical jargon, called *minimal response*), as well as nonverbal confirmations such as head nods. The essential thing to remember is that these responses are never substantive confirmations of what was said but rather indicate "Yes, I'm here, and I heard what you said." You're not saying, "Yes, exactly! I agree with that!"—even if you do. After all, you want to hear the other person's opinion, independent of your own. Instead, you can confirm that you heard that something was said.

Active Listening through Active Summarization of Complex Topics

Summaries are crucial for complex topics. Listen to what your counterpart has said and then return the content in your own words as a question, for instance, "Did I understand correctly that it is important to you that...."

On the one hand, this approach shows clearly that you have been listening. On the other hand, you may uncover misunderstandings quickly and easily. This summarization has another advantage: If you "have" to summarize what has been said, you do not have time to stray to the next question in your head. This tactic keeps your attention on the person you're talking to.

Active Listening by Asking Follow-Up Questions

Follow-up questions are another tool of active listening. Many misunderstandings arise between developers and users because users say something developers believe they understood. This misunderstanding often happens when the user uses the same word that means something different to the developer.

To avoid this kind of misunderstanding, inquiries help. Ask for clarification whenever you hear a statement with content that has not yet been clarified. For example, perhaps a user says, “Yes, and then we just always do handoffs.” As a user researcher, you could reply, “Okay, I see. What does a handoff mean to you?” In this way, you ensure you both have the same understanding.

Active Listening through Approachability

By approachability, we mean seeking eye contact with your counterpart, smiling, and adopting an open body posture. To improve rapport, you can adapt to your counterpart’s speech rate, posture, and wording, as long as your counterpart is in a positive or neutral mood. In general, you should convey a pleasant atmosphere. Your counterpart should feel they are in a conversation, not in an interrogation.

Answering Difficult Questions Takes Time

The “more difficult” your questions become—that is, the more you ask about wishes and non-obvious topics—the longer your counterpart will need to answer these questions. A clear rule in this context is to give your conversation partner the necessary time to respond. Don’t repeat the question. Do not rephrase it. Don’t ask if the question was understood. Just keep silent and wait.

New employees at our agency *Custom Interactions* are always instructed during user research to count slowly backward in their heads from 500 and, when they reach 0, to start again from the beginning. Not a single one has gotten further than 485. The silence feels much longer for the questioner than for the asked.

This time is also called “the client’s sacred time” in coaching. Whenever you interrupt this time, thinking starts all over again. You can quickly tell from the outside whether your counterpart is thinking or has not understood the question. If the person is thinking, they usually look slightly past you or seem to be looking inward. The look seems a bit “glassy.” If the person doesn’t understand you, they usually look directly at you, raise their eyebrows, shrug their shoulders, or ask directly. You’ll know. Promise!

Active Listening versus Documentation

To ensure that your attention is focused on your counterpart, we advise documenting interviews using a video or audio recording if you can. In this way, you can listen actively and take care of the evaluation later. If recording is not possible, the second best option is to have another person take care of the documentation while you conduct the interview. If this tactic is also impossible, we recommend documenting using bullet points and completing the documentation of the content from memory directly after the interview.

In Summary: When Do You Listen "Well"?

You listen well when you do the following:

- You show through verbal signals (minimal response) and nonverbal signals (nodding) that you're taking in what is being said.
- You nevertheless do not influence the content of the conversation by reinforcing or preventing individual statements made by the other person.
- Especially in complex situations, you summarize the statements of the counterpart and thus clarify misunderstandings at an early stage.
- You ask for unknown/undefined terms and, therefore, avoid misunderstandings from the outset.
- You give your counterpart the time to think.
- You make your counterpart feel good.

Thus, in general, listening is characterized by the fact that you listen and show it, which we can summarize by describing a helpful mindset you should bring along as a user researcher.

6.3.4 Your Mindset as a User Researcher

If you observe, ask questions, and listen, you have all the tools a good user researcher must have in their toolbox. These tools can be combined into various methods. We'll detail the methods at the appropriate points in the process (see Chapter 8 through Chapter 13).

No matter which method you use later, the idea or the goal with which you do it is essential. As a user researcher, you should see yourself as a scientist. You want to learn, and your user is the expert on that subject. You're a process expert and very good at getting the knowledge out of the person, but only your users know the usage itself.

To be successful as a scientist, you need some basic attitudes:

1. You should be neutral.
2. You should be interested.
3. You should be friendly.

What do we mean by that? As a user researcher, your aim is to uncover "the truth"—specifically the truth of your users. You may have a hunch about what users care about or suspect the reasons behind a particular behavior. But don't go into surveys to confirm your assumptions. A much easier tactic is to prove your assumptions wrong.

For example, if the Healthyfy team tries to disprove that telemedicine works (i.e., contrary to their idea) and finds no evidence for this hypothesis, conversely, there might be something to it. However, if the Healthyfy team gets many reasons against telehealth from users, then more work is still to be done. Notice the difference—neutrality helps you prevent believing the one user surveyed who sees everything the way you do, ignoring the 99 others who express valid concerns.

Therefore, remain neutral and accept the truth from your users—even if the result is not what you would like to hear. Thus, you should not explain, sell, or judge—even if you find this restraint problematic in a situation because the user has misunderstood something.

Always remember: Outside user research, you're not standing next to every user. Whenever you explain, you dilute the results of your user research. Explanations don't get you anywhere. The only thing that gets you further is understanding the root cause of the problem. This rule is true even if the answer doesn't match you and your values. After all, the focus is the user, not you. User research would be unnecessary if everyone had the same values and opinions.

People are generally good lie detectors. In the introduction to this chapter, we showed you why user researchers often have a hard time: The power behind asking the right questions at the right time is not understood. The reason for this ambiguity involves the skills of the people you're dealing with. Everyone has had conversations and continues to have them in everyday life. As we've hopefully been able to show in the last few sections, being a good lie detector doesn't automatically ensure that you're a good user researcher. But it indeed ensures that you can recognize when someone is being dishonest.

And here we can turn to our second point: As a user researcher, you should have a genuine interest in people. If people don't interest you and you don't want anything to do with them, then the people you talk to will notice your disdain. In turn, their motivation to speak with you, devote time and energy to you, and provide you with the answers you're looking for in detail will be shallow. Think carefully about whether you want to do user research. But if you do—have fun with it, be interested and open.

The last important point is friendliness and politeness. You're dealing with strangers who are supposed to tell you personal details. In this context, of course, rapport helps, but basically, you want to make people feel comfortable in contact with you. The rule is that, if you manage to make everyone you've dealt with feel better afterward than before, you're doing everything right. Strive to be like that guest people keep inviting back to the party.

6.4 Design as a Learnable Process

In addition to understanding users through user research, translating user requirements into a design is the second central activity of usability and UX design. Unfortunately, we often see that people do not dare to approach usability and UX design because they "lack creativity and talent," according to their own statements. We want to give you a brief scientific look at the role of innate creativity and talent to encourage you to approach the topic of usability and UX design.

6.4.1 The Role of Creativity

Looking for a definition of *creativity*, you may come across the following:

> *"Creativity is the ability to produce or develop original work, theories, techniques, or thoughts. A creative individual typically displays originality, imagination, and expressiveness." (American Psychological Association, n.d., "Creativity")*

According to Guilford (1950), creativity has as much to do with bringing known content into new contexts as it is about creating new content. Creativity is not necessarily about pure creation, but partly "only" the rethinking of an existing thing.

In everyday understanding, however, the idea often prevails that a person is either a "creative individual" or not. Creativity is, therefore, innate. If you don't have it, you're on the losing side—at least in the area of usability and UX design. Then, it won't be a good design.

Remember the terms *growth mindset* and *fixed mindset* from the beginning of this chapter? We explained that the assumption that you can't learn something is part of a fixed mindset. With a growth mindset, on the other hand, you assume that you can learn creativity.

Creativity can be trained, and scientists are now quite unanimous that creativity can be specifically learned and increased. For example, some studies show you can increase your creativity by listening to happy music (Ritter & Ferguson, 2017). Twilight, boredom, and personal openness are just as helpful in fostering creative ideas (Stangl, 2021).

6.4.2 The Role of Talent, Practice, Experience, and Knowledge

Like with creativity, many have same with the view of talent: Either you have talent, or you don't.

But dissenting voices have been raised as well: Ericsson, Krampe, and Tesch-Roemer (1993) addressed the question of whether talent or practice is more decisive for success in a particular field. The study became widely known because Malcolm Gladwell mentioned it in his book *Outliers: The Story of Success* (2009), where it was reduced to the simplified statement that 10,000 hours of practice in a field will make you a master.

In the meantime, research (i.e., Macnamara & Maitra, 2019) has proven that practice, while a crucial prerequisite for success, is not sufficient alone for top performance. You'll see a mix of talent, practice, and environment explaining the differences between very good or outstanding individuals in the field. Individuals often find it easier to invest time in practice if they have a particular talent level in an area. If you're initially good at something, you'll achieve a quicker sense of accomplishment. This quick success motivates you to practice more, resulting in even more feelings of success. As a result, you keep practicing and get better bit by bit.

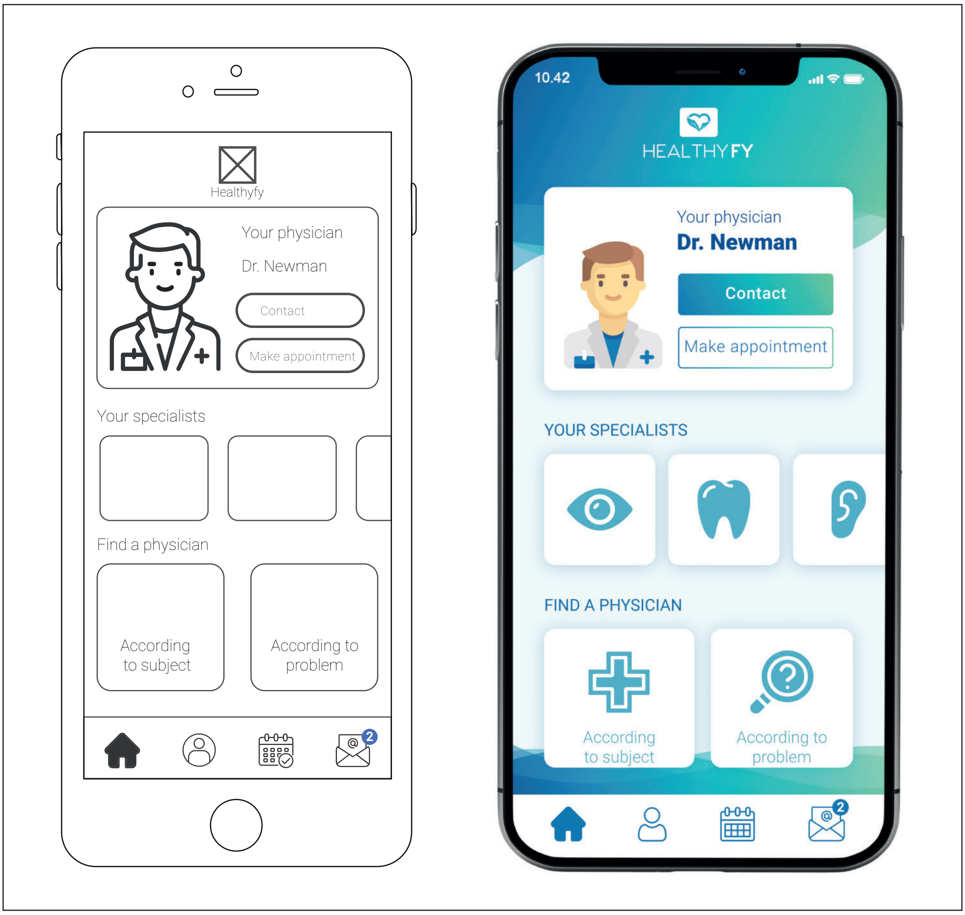

Healthyfy
Your physician
Dr. Newman
Contact
Make appointment
Your specialists
Find a physician
According to subject
According to problem
2
10.42
HEALTHYFY
Your physician
Dr. Newman
Contact
Make appointment
YOUR SPECIALISTS
FIND A PHYSICIAN
According to subject
According to problem
2

Figure 6.7 A wireframe is shown on the left, and a UI that a visual designer has already worked on is shown on the right.

Does this mean you must have been an artistic genius as a child? No, it doesn't, for the following reasons:

- Research has shown that practice always helps improve skills. No master is born a master. Masters invested a lot of time until they became masters. Other people who invested a similar amount of time but had "no" talent were "only" very good or at "worst" good in the study. However, they were always much better than individuals who invested little or no time. Moreover, the "master" requirement is rarely necessary in practice. You can achieve outstanding usability and UX even as an amateur.

- Usability and UX design requires much more than what is often understood by "design"—making things pretty. Design also includes, for example, the conception of a UI in the form of wireframes. (More detailed information follows in Chapter 11. However, the mockup shown in Figure 6.7, on the left, gives you a first impression.)

  Usability and UX design requires different design skills than, for example, only visual design (i.e., actually "making things pretty," as shown in Figure 6.7, on the right). Thus, if one design area doesn't fit, others involve different requirements.
- This book suggests you consider usability and UX design as a process. We provide concrete rules and assistance that "only" must be applied. This approach requires less innate talent but rather emphasizes practice and process fidelity.

We assume you'll not be influenced by the "talent stamp." If you enjoy the creative activity, then practice it and get better.

6.4.3 What Do I Need to Learn to Become a Usability and UX Designer?

The field of usability and UX design is no longer brand new, but few degree programs focus on these areas even now. The backgrounds of the experts we encounter in practice are correspondingly diverse. The following backgrounds can help you do successful usability and UX design:

- Communication design
- Graphic design
- Interactive media design
- Software development
- Engineering
- Psychology
- Communication studies

This list is not exhaustive but represents only a selection of the possible and common starting points. However, we also know linguists, philosophers, business economists, and others who are successful usability and UX designers. Also, usability and UX design is not exclusively reserved for those who get a degree; you can also build your skills through online graphic design courses and freelancing work. In general, if you're interested in and practice a particular subject, you'll get better at it.

Each of these professional backgrounds has a different focus and imparts different strengths. The design-heavy fields have a clear advantage in finding creative solutions and creating design systems. The more human-centered fields have an advantage in understanding users' mental models. The communication-related areas have an advantage in designing text, captions, and, more generally, UX writing. Engineers have a very good understanding of structure, order, and technology. People from software development have a starting advantage in developing design systems because they understand how implementation happens later.

6.5 Testing and Corporate Culture

The interplay between user research and usability and UX design is most evident in the testing of solutions. With testing, both areas of expertise come together and play to their strengths: Researchers find out what fits or doesn't. Usability and UX designers translate the findings into better solutions.

Nevertheless, many of our inexperienced customers initially shy away from subjecting their products to user testing. In addition to the effort required for user testing, fear exists in most cases. Imagine you test the product, and the feedback is all bad. Then, you have to justify why *you* did a lousy job. We often encounter this fear. At this point, we have two arguments against succumbing to this fear of testing:

- Not testing means not knowing how your product stands. But that doesn't change the quality of the product. If the product is terrible and untested, the product will stay flawed. Something can be changed if the product is terrible, but the reason becomes evident in a user test. The fundamental mistake is not having a bad product but closing your mind to opportunities for improvement.
- The quality of the product comes to light at some point anyway. At the latest, the pressure to justify increases with poor sales figures or lousy customer reviews. So, if you don't give notice quickly beforehand, you can't avoid the discussion. From our point of view, quality is better to address as early as possible—because then changes are more favorable.

Overall, the discussion of fear and error strongly indicates a fixed mindset in a corporate culture. However, many superiors and companies are now at least familiar with the idea that user tests should be performed and are often much more open to the topic than the developers themselves. This imbalance is not surprising since user tests with the associated growth mindset do not lead to a downward spiral, but an upward one.

Instead of:

> *Testing = Bad feedback = I failed = I can't develop good products = More fear of failure = failure → Testing = Failure (Fixed Mindset)*

Opt for:

> *Testing = Bad feedback = I now know what I can do better = The product is getting better = More success → Testing = More success (Growth Mindset)*

If you're like most people, the term "success" has a positive connotation for you. Therefore, remember the growth mindset and the associated formula *Testing = More Success*.

If your company is afraid of getting into trouble due to poor results in user tests, we would like to provide a few tips on precautionary measures:

- At the beginning of development, make it clear that you want to schedule tests with users to obtain feedback on the current status in good time. You can also refer to the literature sources from Chapter 3 to justify your decision.
- Try to propose a process where you can test as early as possible with as little cost as possible to show after a short time that you're on the right track with your work. Here, repetitive tests are especially important, which deliver better results step by step.
- Make the results of the tests accessible. Seeing or hearing helps better than a written summary. For example, use highlight videos or audio snippets to show what works and what doesn't.
- Invite decision-makers to participate in the testing. If necessary, find an opportunity to test an existing system. This activity increases awareness of the current status and the problems encountered by users.

We hope you can use these tips to increase enthusiasm for user testing in your company and make your everyday life easier. Have fun with testing!

Chapter 7

Foundations

"An investment in knowledge pays the best interest."
—Benjamin Franklin (statesman, publisher, writer, natural scientist, and inventor of many things, like the lightning rod)

In this chapter, we'll introduce you to the fundamentals of psychology and design that are important for a deep understanding of the data-driven user experience (UX) design process (3DUX) in later chapters. Reading this chapter will be worthwhile. We know foundation chapters can be exhausting, and we have tried to do the opposite.

First, Section 7.1 through Section 7.3 focus on human perception, memory, and information processing. In many respects, these topics form the basis of interactions between humans and machines. If you understand this basis, the interface design is more straightforward.

We deal with design basics in the second part of this foundational chapter (in Section 7.4). We explain the general effect of design through images, shapes and colors, design laws, and heuristics.

7.1 Psychological Foundations: Don't Worry, It's about People

Even though the book has the word "design" in the title, we firmly believe that psychological know-how is essential for good usability and UX design. But maybe that's a personal bias; Michaela is a psychologist.

But seriously, how will you bring good design to life if you don't understand how people perceive, remember, and decide? Products are solutions to problems. Problems are people's problems. So, if you want your products to solve people's problems, you must understand how people "work."

We are, of course, aware that every person is unique. However, some principles work the same or at least similarly for most people. We want to familiarize you with these principles in this chapter.

7.2 Perception: The Human Sensory System

The human sensory system is the first area we'll focus on. This system includes all the sensory channels through which humans can receive information from the external world. A person would be cut off from the outside world without this sensory system.

In the book *Wie Design wirkt* (*How Design Works*) by Monika Heimann and Michael Schütz (2017), a human is compared to a submarine. A human is a closed system that only establishes contact with the outside world through sensor technology. We find this comparison extremely apt. A human being can only perceive what they have a suitable sensor for. In humans, the sensors are called *sensory organs*. With the help of the various sensory organs, people collect impressions from the environment and process them further in the brain.

Understanding perception is essential for your work in usability and UX design to bear fruit. Only when your design gets attention (and is thus discovered and, in the best case, recognized) is there a chance that your design will have an effect. However, as Heimann and Schütz noted, attention does not equal impact. *Attention* first means the realization that your design has been discovered. The *effect* describes what happens afterward with your design in the user's mind. Therefore, perception does not occur in the design but in the minds of viewers.

Example: Perception Arises in the Mind of the Observer

Let's consider an example from *Wie Design wirkt* (*How Design Works*) by Monika Heimann and Michael Schütz. Projectors work by throwing light onto a white wall or white surface. Depending on the color to be shown, the light is white or colored. Do you know what color the light is in the places where the image is black? None at all. No light is thrown at all. Not spectacular in itself.

What is spectacular, however, is that the same surface you perceive as a black surface in a projector image, you would identify as a white wall when the projector is switched off. The color of the wall does not change, of course—and the projector did not cast any "black light" on the wall. Your white wall is, therefore, the darkest color that the projector can display. Perception is fascinating, isn't it?

The more your design can tie in with experiences, memories, and emotions, the more associations it triggers. Therefore, let's now look at the individual sensory systems of humans as an overview of the available sensory channels and their capacities before we look at the fundamentals of design. In this way, you can consciously decide which design your system, product, or service should fit the perceptual possibilities best.

To start, humans are fantastic at perceiving information. Perception is taken for granted by most of us, and the masterpiece behind information processing remains invisible to most people. Yet conscious perception is a gigantic feat. The volume of sensory impressions that affect us every second is enormous.

In her essay on consciousness research, Michelle Kreisl (2013) at *Austria-Forum.org* states that about 11 million sensory impressions are processed in the brain every second—an almost unbelievable quantity. However, only about 40 of these impressions are consciously processed. The brain of a human being is, therefore, extremely good at blocking out sensory impressions.

But where do 11 million sensory impressions per second come from anyway? These sensory impressions come from the totality of all sensory systems that a human being has at their disposal. Even if individual sensory impressions predominate in our culture—vision is perceived as the primary sense—a person has various sensory channels through which they perceive information from the environment. But first things first. Let's start with the eyes.

Regarding anatomy and physical relationships, we resort to simplified explanations to facilitate a basic understanding to help you design solutions. To learn more about the subject, we recommend *Sensation and Perception* by Goldstein and Cacciamani (2021). In that book, you'll learn, in almost 500 pages, how human perception works in detail—without direct reference to design or UX. Much of the content we present in this chapter is based on this highly comprehensive standard work.

7.2.1 The Eye

The probably most apparent sensory system in humans is the eyes. The eyes process light stimuli and enable us, for example, to perceive information at a great distance or in a written language. The importance of eyesight in our society and daily life is shown, among other things, by the fact that blindness counts as a severe disability. Behind this view of blindness is the assumption that the ability to see is fundamentally essential for everyday interaction.

The eyes are a powerful tool in this process. They can perceive colors, movement, and depth, thus conveying information with very high information density.

The Components of the Eye, Accommodation, and Fixation

The eye is relatively complex and has many components, as shown in Figure 7.1. Although all parts of the human eye are relevant, for this book, we'll focus on the parts of the eye that are important for designing user interfaces (UIs).

From our point of view, these areas are the pupil, the lens and its associated ciliary muscle, the retina, and the macula.

The pupil is the small opening (the black dot) in the center of the colored iris. Light from the environment enters the eye through the pupil. The pupil's size—its width—adjusts to the ambient brightness. When the environment is bright, the pupil is small. When it is darker, the pupil dilates to let more light into the eye. In addition to the ambient brightness, emotions such as fear, excitement, or joy can also cause the pupil to dilate.

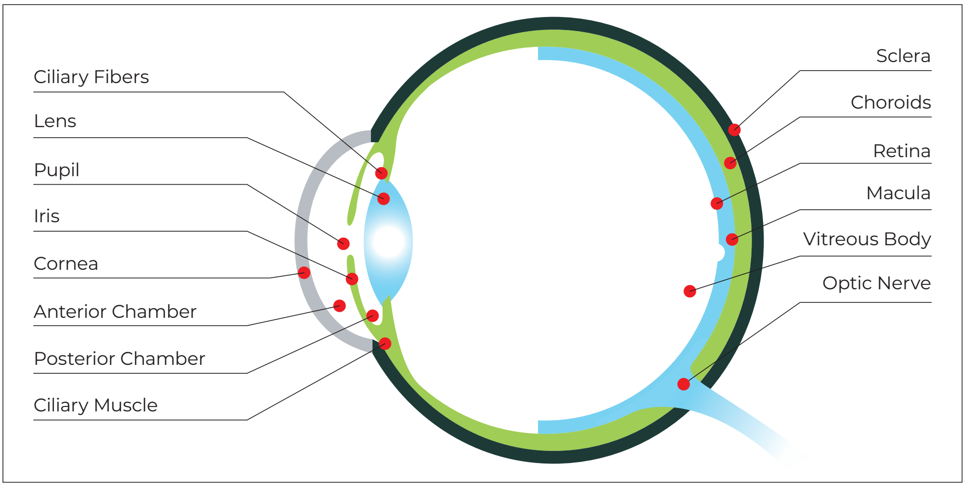


Figure 7.1 Structure of the human eye.

The lens at the front of the eye focuses the light entering through the pupil and throws it onto the retina. The lens is elastic and can be compressed by the ciliary muscle that surrounds the lens. The compression changes the refractive power of the lens, allowing the eye to focus at different distances, enabling you to see both near and far objects in focus. The ability to adjust focus to distance is called *accommodation*. Unfortunately, the elasticity of the lens usually decreases with age. You may even notice this in yourself when typical age-related long-sightedness develops.

The retina lines the eye from the inside and is the actual sensor of the eye. It contains approximately 126,000,000 light receptors that convert incoming light into nerve impulses and send this information directly to the brain via the optic nerve. We'll take a closer look at these light receptors in a moment.

The final essential component of the eye is the *macula* (sometimes called the "yellow spot"). In the retina's center, the macula takes its name from the dye lutein, which is stored here and makes the spot appear yellow. In the center of the yellow spot, there is a small dent or depression called the *fovea centralis*, which is the site of the sharpest vision in the eye.

The fovea centralis has more light receptors than any other part of the retina. This area enables us humans to see a specific point exceptionally sharply. When you focus on an object with your eyes, your eyes automatically adjust themselves so that the object is imaged on the retina in the fovea centralis. This ability to adjust also shows how well our brain works. Most people who have never dealt with this aspect of perception feel that they can see sharply everywhere. However, the angle is tiny, and the sharp image you think you see is a masterstroke of your brain. Try it out for yourself: Focus on the point shown in the center of Figure 7.2 and read the words around it without moving your eyes away from the dot.

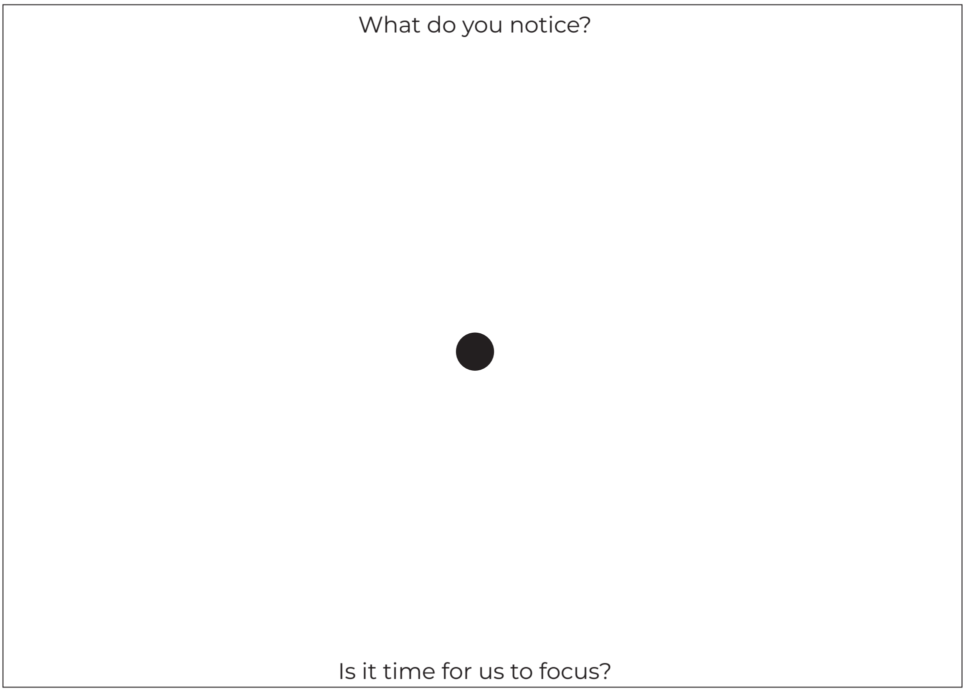

What do you notice?
Is it time for us to focus?

Figure 7.2 Image demonstrating sharp vision.

What do you notice? Most likely, you can't read the text, or it's blurry. You may also have scanned the text in passing. Unfortunately, you cannot see sharply outside the fovea centralis. The sharp overall image is calculated in your brain, so to speak, by putting together various individual images that are sharp. If you ever observe people's gaze, you'll see that the gaze often wanders and rarely stays in the same place for a long time. In this way, your brain develops the overall image you see as sharp. Seeing sharply at one point is one of the core tasks of the eye and is called *fixation*.

But even if you only fixate on one spot, you still see peripherally on a much larger area. You can perceive information as changes in the area outside of sharp vision. Therefore, with the eye, many pieces of information can be perceived in parallel.

Additionally, you can consciously decide to turn your attention to another aspect, which makes the eye a what's called a *simultaneous sense* (Heimann & Schütz, 2017). Pretty handy if you ask us.

The brain's performance combined with the eye becomes even more fascinating when we look at *color vision* in more detail. In fixation, we had already established that the brain calculates the perceived sharp overall image from many images. The same is also valid for color vision. You see in color only in a small part of your eye.

In the fovea centralis, the density of the light receptors is exceptionally high. You'll also find a particular type of light receptor there, namely, the cones. *Cones* are attuned to produce images as detailed and as accurately as possible in daylight. To accomplish this task, three different types of cones convert light in different wavelengths into nerve stimuli. The cones are called S, M, and L (after *short*, *middle,* and *long*), referring to the wavelength of light processed. The S cones perceive the blue light range; the M cones, the green light range; and the L cones, the range of red light. As shown in Figure 7.3, the visible light spectrum represents only a tiny part of the available wavelengths. Many kinds of radiation are not perceived by our eyes but can be detected by other animals (e.g., bats) or by technical devices.

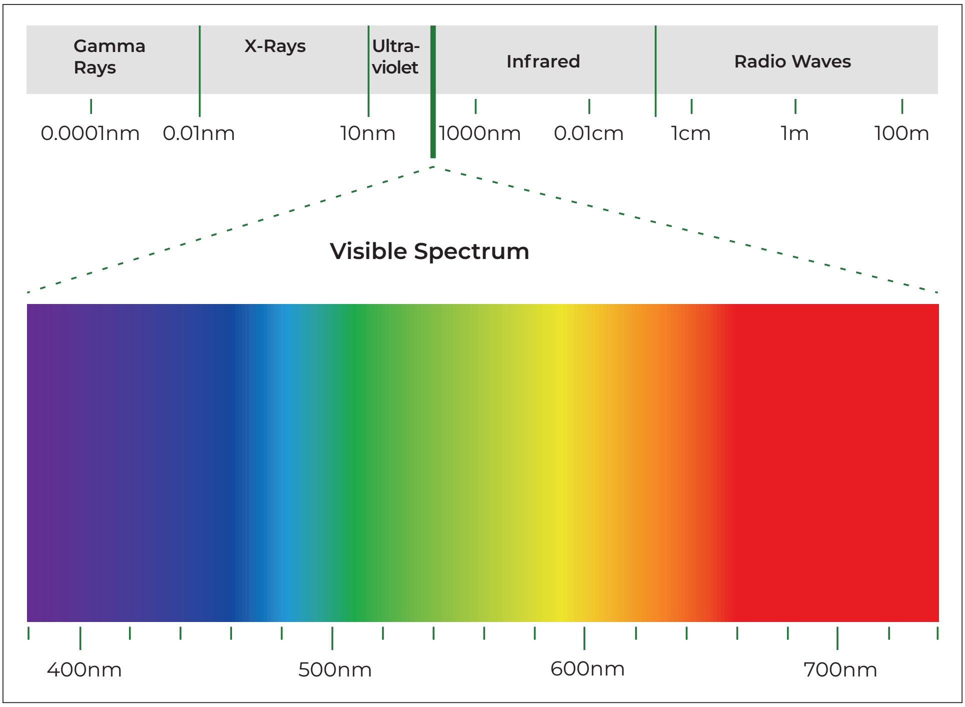


Figure 7.3 Light waves, including the visible spectrum.

The fovea centralis exclusively contains cones, as shown in Figure 7.4. Outside the fovea centralis, rods predominate. In total, your eye has significantly more rods than cones, reportedly 120,000,000 rods to only 6,000,000 cones (Goldstein & Cacciamani, 2021). Nevertheless, cones are found throughout the entire retina.

But now to the rods, as shown in Figure 7.4: *Rods* are light sensors that respond to even the most minor amounts of light, allowing us to see even in dim light. In contrast to the cones, rods cannot distinguish between different colors, only between light and no light. As a result, outside the fovea centralis, you see in black and white. This loss of colors is shown in Figure 7.5.

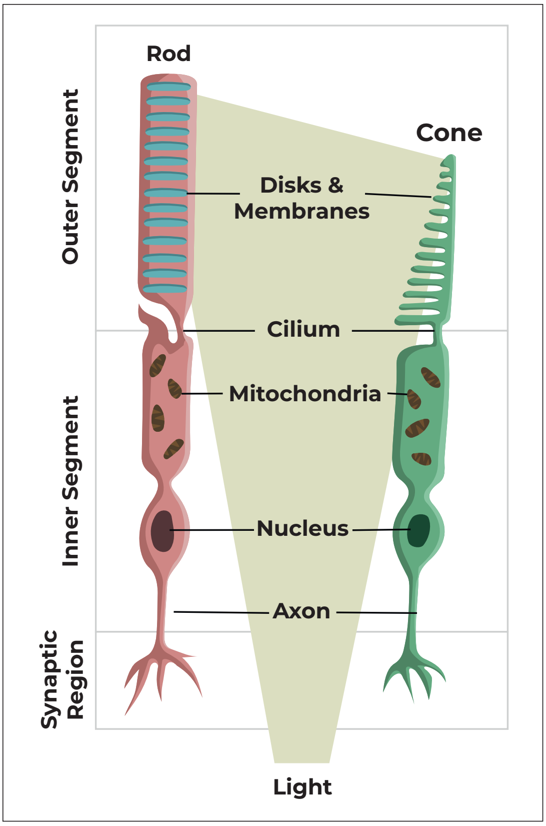

Rod
Cone
Outer Segment
Disks & Membranes
Cilium
Inner Segment
Mitochondria
Nucleus
Axon
Synaptic Region
Light

Figure 7.4 Cones and rods. (Source: Modified from Systemoff/Shutterstock)

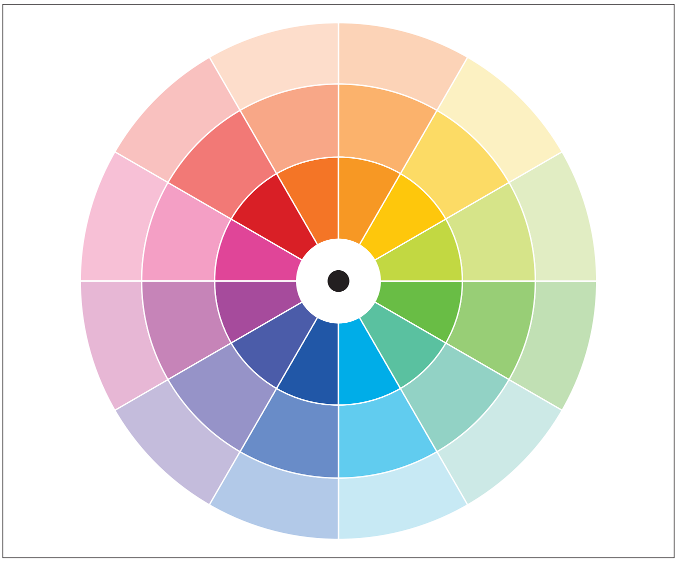


Figure 7.5 Illustration of color vision outside the fovea centralis.

Light-Dark Adaptation

At this point, you've learned about the main components of the eye for perception and, with accommodation, one of the essential functions of the eye (focusing on objects at different distances). But the eye can do much more!

Another function of the eye and its sensory cells is *light-dark adaptation*. This term refers to a process in which your eye adapts to different levels of brightness. Perhaps you know the feeling when sitting in a dark movie theater and suddenly the light comes on or when you drive your car into a tunnel from bright sunshine. At first, recognizing anything in the new lighting situation is difficult.

When you come into the light, your eyes quickly get used to the new lighting situation, and you can see at least something quickly. However, if you move from a bright environment to a dark one, the process takes a moment. About 3 to 4 minutes will be required for the cones to adapt to the dark environment and allow you to see reasonably well. Thus, when you're in a tunnel, you're usually asked to turn on your headlights even though the tunnel is lit. In fact, about 25 minutes is required for your eyes to achieve the best visual result in the dark. That's how long it takes for the rods to "recover" from the light. But when they do, their sensitivity to light is about 100,000 times greater than in the light state. Thus, if you wake up in a dark room, you can still recognize many details and even see shadows on the walls, even though when you go to bed, the room seems pitch black at first.

Now, with light-dark adaptation, accommodation, and fixation, you know the three core abilities of the eye that allow us to do so much in the area of vision. For you, this knowledge is vital for several reasons:

1. You now know that people can absorb information in one place when it needs to be read.
2. You know that the eye must work to adapt to different visual distances and that the ability to adapt decreases with age.
3. You know that it takes the eye a while to switch between light and dark environments and that adjusting from light to dark, in particular, is time consuming.

Color Perception and Color Vision Deficiency

All these findings should already help you develop better products. But you know even more: We have already discussed the three color perception cones. Now, you might falter and ask yourself, "But I can see more than just three colors, can't I?"

Yes, of course, you're right! It is believed that the human eye can distinguish about 200 different colors through the interaction of the cones. Combined with different intensities, these colors can be perceived as brighter or darker. By adding white, saturation can be adjusted. In this way, about 500 brightness levels alone can be distinguished. The human eye is assumed to perceive approximately 2.3 million color tones (Wäger, 2017).

We know that your "design heart" is beating faster now, and you'll probably immediately think of the many possibilities that this variety opens up. Let's slow down a bit at this point.

The human eye can detect differences of approximately 0.5% in adjacent surfaces (Wäger, 2017), which is equal about 200 levels from 0% (black) to 100% (white). Thus, the 256 levels used by digital black-and-white images is usually more than adequate. More colors are, therefore, not necessarily needed.

In addition, with the *relative perceptual capability*, if you place different hues or brightnesses next to each other, you can distinguish the 200 or even 500 brightnesses. You should not use 200 or 500 brightnesses in different places in your product and trust that they will be perceived as distinct (which would be an *absolute perceptual capability*). Unfortunately, your relative perceptual capability is much lower.

Maybe you even know the phenomenon yourself. Recently, we wanted to book a vacation at a well-known resort chain in Germany. In this case, we could determine the location of our desired vacation home using a color-coded map. Unfortunately, the colors were sometimes rather similar, and the legend is far from the houses, so we could only orient ourselves by the numbers.

Still, it didn't stop us. As a usability and UX designer, you must use colors clearly when designing your solutions. In addition, you should refrain from coding information exclusively via color as much as possible. Why?

You have already learned that color vision results from the interaction of the different cones. However, these cones can be damaged or missing, limiting the ability to perceive color. This limitation is called *color vision deficiency*. Depending on how many cones are damaged or missing, a person might not see in color at all, only in black and white or gray, or can only see specific colors.

The incidence of *complete color blindness* is infrequent, at 1 in 100,000 people. *Color vision deficiency*—the absence of certain cones—on the other hand, is relatively common. Depending on the source, the overall figures vary between 2% and 10% of men and less than 1% of women. The big difference between men and women stems from color vision deficiency often being inherited via the X chromosome. In this case, women are clearly at an advantage since, with two X chromosomes, they still have a reserve up their sleeve. However, one in about every twenty men perceives colors differently.

The most common color vision deficiencies are called *protanopia*, *deuteranopia*, and *tritanopia*, as shown in Figure 7.6. In the first two forms, the affected person sees only shades of blue, gray, and yellow. The color image of both color vision deficiencies is quite similar, but with a different point where the blue turns gray. Only blue, gray, and red tones are perceived in tritanopia, although this color vision deficiency is rare.

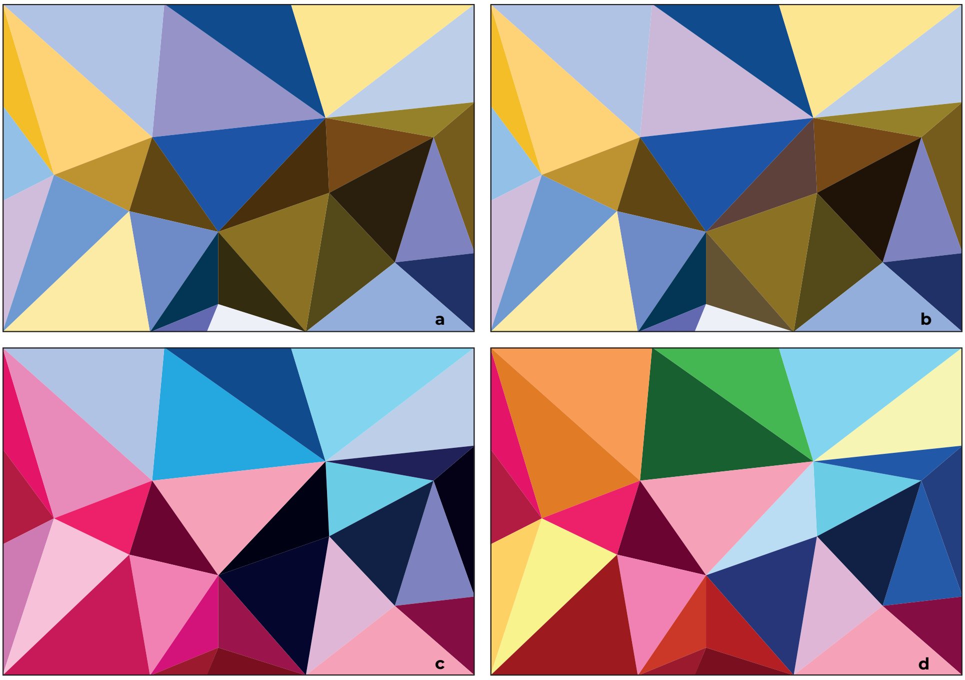


Figure 7.6 The three most common color vision defects are protanopia (a), deuteranopia (b), and tritanopia (c), compared to the original image (d).

You should take this knowledge into account when designing your products. Especially the popular red-green coding to show whether everything is okay quickly reaches its limits here. Understanding perception basics will also help. Therefore, we recommend combine color coding with shape coding, as shown in Figure 7.7. Using both approaches will make your design much easier to recognize—even for people who do not have color vision impairment.

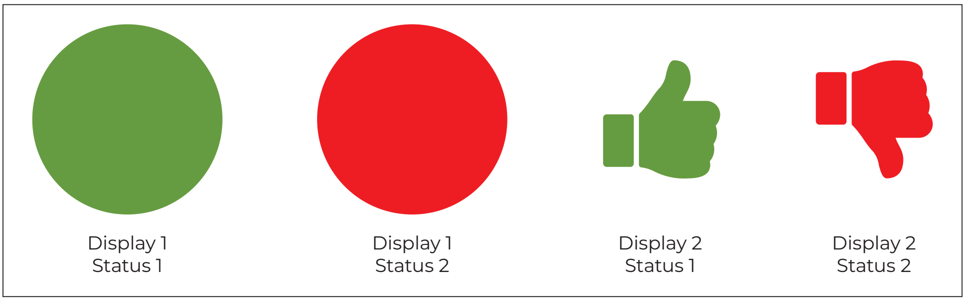


Figure 7.7 On the left (display 1), the information “good” or “bad” is only coded via the color. On the right (display 2), the information is additionally coded via the shape (thumbs up/down).

Depth Perception

As we've seen, our eyes are absolute marvels. They enable us to perceive various impressions through accommodation, fixation, adaptation, and color perception. Another essential ability is *depth perception*, which allows us to orient ourselves in three-dimensional space.

The brain needs information from at least one eye to make simple decisions—what's called *monocular depth cues*. For example, the size of an object is used to derive an estimate of distance. If the object's size is known (e.g., an adult human), then this perceived size can be used to decide how close this person is to the observer. This depth cue is also called *familiar size*.

The *relative height in the field of view* is also a monocular depth criterion. The higher objects are positioned in the field of view (towards the horizon), the further away these objects usually are from the observer.

On the other hand, the *relative size* indicates that, with two identical objects, the one further away is perceived as smaller.

In addition to size and height information, the brain also uses *convergence* (of lines at a distance) to indicate depth. If you ever stand on a road and follow the road to the horizon with your gaze, you'll see that the road—although the same width—seems to converge at the end. The whole thing is called *perspective convergence* and is also used to indicate distance.

Additional monocular depth cues may arise due to atmospheric perspective. For example, farther away objects are often a bit blurrier and slightly bluish than closer objects.

Also, the perceived density of objects—what's called *texture gradient*—allows conclusions to be drawn about distance. For example, if you look at a field of flowers, it seems as if the flowers are closer together as the distance increases.

The *occlusion of an object* also gives clues about which object is in front and which is behind.

The last monocular depth cues are *shadows*. Depending on the position of the shadows, much can be seen about the position of the object or its height.

This whole set of monocular depth cues explains why a person can orient themselves well in three-dimensional space even with limited perception (only one eye). In addition to monocular depth cues, you perceive space through what are called *motion-induced depth cues*. These cues include the detection or nondetection of objects and the perception of occlusion—but in motion. If one object moves past the other, then the object that can be seen completely is recognized as the one in front, and the (partially) hidden object, as the one behind.

Another motion-induced criterion is called *motion parallax*. Imagine that you're sitting on a moving train and looking out of the window. All objects close to you (bushes, trees) seem to rush past you. On the other hand, all things at a greater distance do not seem to move at all or only move slowly.

Only now—after you have already perceived the distance and depth in the most diverse ways—does the second eye come into play. Although even one eye is a marvel in itself, two eyes together are worlds better. For example, you're not allowed to drive a truck or bus in Germany if you can only see with one eye. In the US, you must submit to medical checks and may have restrictions and other requirements on your driver's license (Fowler, 2021).

So, let's look at *binocular (two-eye) depth cues*. The core advantage of seeing with two eyes is that each eye "receives" its own images, and the differences between the images can be used to calculate distance.

Two Eyes, Two Different Images

Don't believe us that you see something different in each eye? Here's an easy way to see the difference between the two images: stretch your arm straight forward and clench your hand into a fist. But keep your thumb pointing upward, as in a "Super!" gesture. Now, close your left and right eyes alternately. What do you notice?

Your thumb seems to jump back and forth. This jump is because your left eye sees the thumb slightly further to the left, and your right eye sees the thumb slightly further to the right. The jump would be much smaller if your arm were 10 feet long. If you hold your hand 10 inches in front of your eyes, the jump will be even more significant (try it!). The difference between the left and the right retinal image is called *binocular disparity*.

Based on the eye position and the deviation of the two images from each other, your brain can decide in a flash which object is how far away and which object of several objects is further ahead or further behind. This determination happens while we are usually unaware that we have different retinal images. Unbelievable!

For you, knowledge of depth perception in design is relevant so that you can create the desired impression via the design. In most cases, the medium with which you design digital products and through which the products are perceived is two-dimensional. Whether for a PC, TV, smartwatch, or smartphone, the result is a two-dimensional image. If you want to breathe the feeling of "life" or "depth" into this image, you can think and work in two- or three-dimensional space. After this excursion into depth perception, doing this is quite simple, as shown in Figure 7.8.

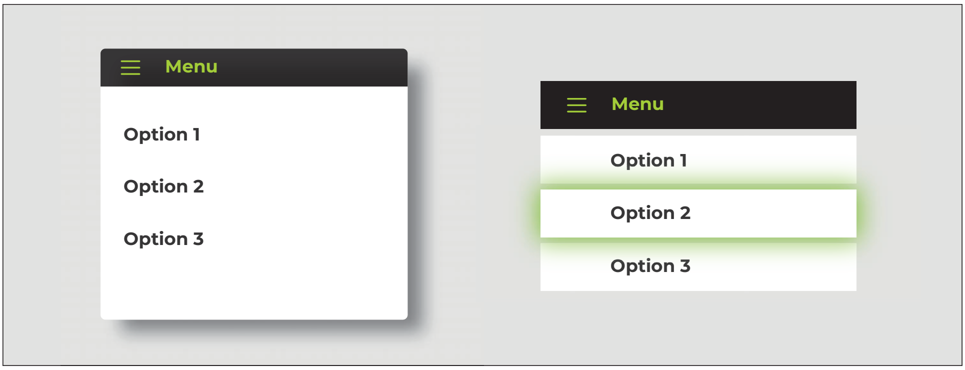


Figure 7.8 Various digital UIs that use depth perception criteria to create a depth effect.

Key Takeaways for UI Design

We have written a lot about the properties of the eye in this section. In part, we have also drawn initial conclusions for the design, but we would like to go into these concepts in more depth in this section.

When designing visual displays, make sure that users are not exposed to alternating bright and dark operating views. Since the eye must adapt to the changed brightness each time, time is lost unnecessarily. If the differences are significant, users will also be downright dazzled. You may experience this when you use your smartphone in the morning while still half asleep and switch to a relatively bright app (e.g., checking your email). Actual glare may make your eyes start to water.

Avoiding light and dark areas applies from one view to the next and within a view. Thus, you should also make sure that very dark areas are not used in combination with very light ones.

This advice also applies to the environment in which your operating display will be used. If you design a display for a bright environment, you should also use bright surfaces. If users use your display in the dark, use dark colors for the large areas.

An example of a UI in which both design forms occur is the navigation system in a car. Typically, the system has a day view with light areas and a night mode with dark areas, as shown in Figure 7.9. This setting ensures the light-dark adaptation is less pronounced and no glare occurs at night. The often popular *dark mode* (the design with dark areas) is not well suited, however, to a bright environment. If users nevertheless desire using the “wrong” mode, this choice is more likely a matter of personal taste in visual design.

To capture a user’s attention through visual design, take advantage of the fact that peripheral vision responds primarily to change. For example, a warning designed visually should therefore flash. The resulting permanent change makes it much easier to perceive than a static display. Depending on the criticality, you should also consider warning colors (yellow or red) as well as the combination with acoustic indications (Section 7.2.2). A practical use case would be, for example, the design of a control center—that is, a large display area on which users must view and monitor a lot of information.

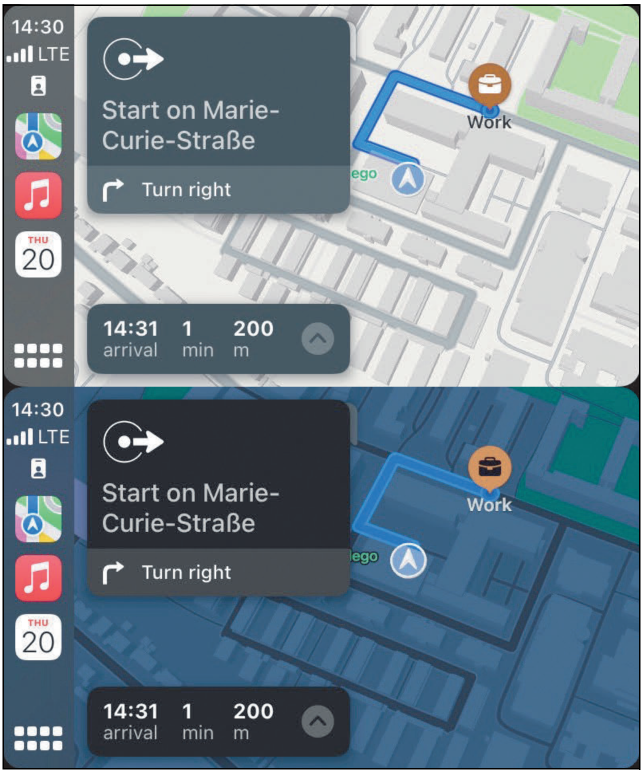

14:30
LTE
THU
20
Start on Marie-Curie-Straße
Turn right
Work
ego
14:31 arrival
1 min
200 m
14:30
LTE
THU
20
Start on Marie-Curie-Straße
Turn right
Work
ego
14:31 arrival
1 min
200 m

Figure 7.9 Day mode (top) and night mode (bottom) of a navigation system (Apple CarPlay).

The *accommodation* area—that is, focusing on objects at different distances—also has some influence on the design of UIs. Since switching between near and far takes time, and the time required increases with age, you should avoid these changes in distance in the design if possible.

A practical example is driving a car: In this case, accommodation often occurs in the near range (speedometer, infotainment system) and the far range. Ideally, you can design without this difference and, for example, install a head-up display allowing the user to keep an eye on the road and the distance. If not possible for cost reasons, for example, then you can help your users by enabling quick orientation. For example, speed control can be supported with a large speedometer in which frequently driven speed ranges are marked. Users can then quickly check, for instance, whether they are moving in the range of up to 25 miles an hour. In this way, drivers can know whether they are driving too fast in school zones, even without reading off the exact value.

In our earlier section on *color perception*, we introduced you to various color misperceptions. We noted that you should not transport information exclusively via color. A best practice is to use a combination of color and shape to ensure that perception is simple, which not only benefits people with color vision deficiencies but also helps everyone.

From the eye's characteristics in color perception, it also follows that you must pay attention to *sufficient contrasts* in the design if people are to recognize distinct elements. If, for example, different areas are to be marked, you should use colors with solid contrast to each other. A red and a blue area can usually be distinguished better than a blue and a light blue one. Since color vision deficiencies can also play a role in this context, we recommend that you ideally also code using the shape.

As mentioned earlier, you can use a *flashing display* to attract attention. However, to draw attention, you can also use *depth perception*. For example, occlusion ensures that the occluded object is perceived as further away. So, in a UI, you can put one window in front of another to show that people should deal with the front window first. This approach is common, for example, with an error message or a popup window. If you want to draw even more focus to the element in the foreground, you can also blur the background to create a depth effect.

You see: The eye is versatile, and thus, many design options are available. Since, in addition, combinations with other sensory organs often play an essential role, let's now turn to the next sensory wonder: the ear.

7.2.2 The Ear

The ear is the second most crucial sensory channel for most people. In contrast to the eye, the ear is not a simultaneous sense but a *successive sense* (see Heimann & Schütz, 2017). A virtually impossible task is to follow two conversations simultaneously and perceive both meaningfully. Also tricky is consciously switching between two equally loud audio sources because you perceive the sounds from both audio sources.

However, we can also consider these features an advantage: Even if you focus on one audio source, a second one can provide, for example, an audible warning that you can perceive (assuming sufficient volume). This situation is different from the eye, where you must look in the direction of the signal source, and the signal must additionally be noticeable enough.

But first, let's provide an overview of the ear's structure before going into the specifics of acoustic perception.

The Components of the Ear

The ear is also a design masterpiece capable of recording a wide range of information for us. Only through our ears, for example, can we identify the direction of sounds and communicate linguistically, even without visual contact. The ears enable us to perceive danger at a greater distance and help with orientation. To do these tasks, the ear needs different parts. In general, the structure of the ear is differentiated into the following components, as shown in Figure 7.10:

- The outer ear
- The middle ear
- The inner ear

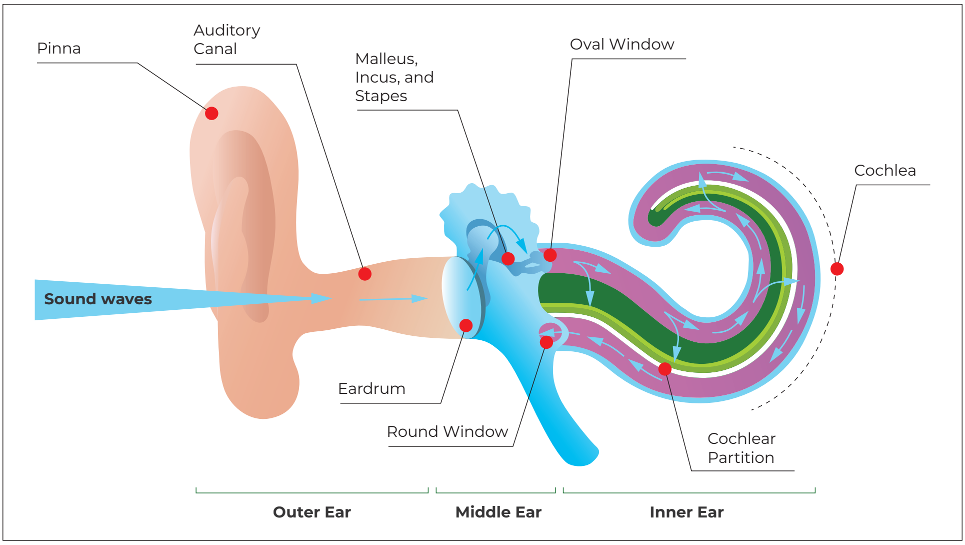


Figure 7.10 Representation of the ear with the outer, middle, and inner ear. (Source: Modified from VectorMine/Shutterstock)

The *outer ear* is what you can see from the outside. It includes the pinna and the external auditory canal leading to the middle ear. For acoustic perception, the outer ear is only of minor importance. Also, the absence of the outer ear does not cause you to become deaf. However, the pinna is especially important for sound localization because its shape allows sound waves to be transmitted from different directions with different frequency patterns.

The *external auditory canal* adjoins the pinna. This component protects the more internal parts of the ear from injury and helps maintain a constant temperature. The shape and length of the external auditory canal also mean that specific frequencies are amplified and can therefore be perceived better.

To perceive anything at all, the next part of the ear is centrally important: the *eardrum*. Located about 1.2 inches “behind” the ear opening in adult humans, the eardrum is stimulated to vibrate by incoming sound waves. Not all sound waves are heard equally well. What’s called the *resonance range* lies from 2,000 to 5,000 hertz. Human speech is heard particularly well and perceived more clearly.

To transmit these vibrations, the eardrum passes them on to the structures in the *middle ear.* The structures in the middle ear are small ossicles, transmitting the movement further. In this process, the vibration is concentrated from the “large” area of the eardrum to the much smaller area of the stapes. In addition, the interaction of the various bones (*malleus*, *incus*, and *stapes*) acts like a lever that amplifies the incoming sound.

This translation of sound waves into bone motion is necessary because the outer ear canal is filled with air, but the inner part of the ear is filled with inner ear fluid. Since this fluid has a much higher density than the air, direct transmission is only possible with substantial losses—similar to having your head underwater while trying to understand what is being spoken outside the water (cf. Goldstein & Cacciamani, 2021).

The stapes transmits the vibration to the *cochlea*, as shown in Figure 7.11.

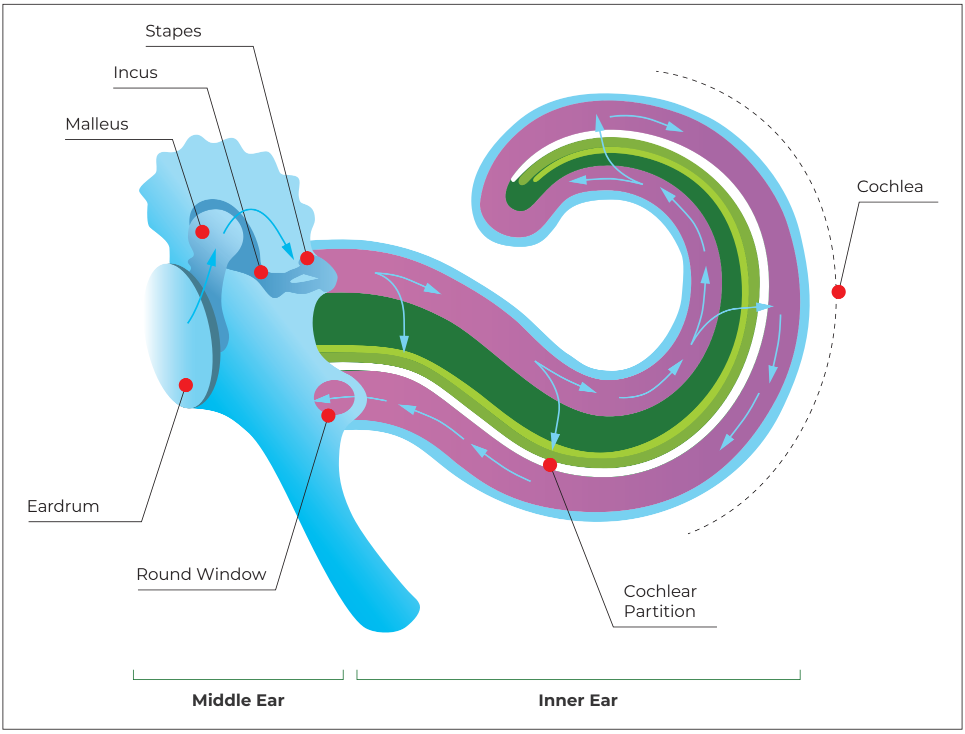


Figure 7.11 The stapes as a transition from the middle ear to the inner ear, including the cochlea. (Source: Modified from VectorMine/Shutterstock)

The *cochlea* contains the inner ear fluid and is separated into two parts by the *cochlear partition.* This partition goes almost through the entire cochlea and contains the receptors for the actual hearing. In this case, the receptors are not cones or rods but hair cells, as shown in Figure 7.12.

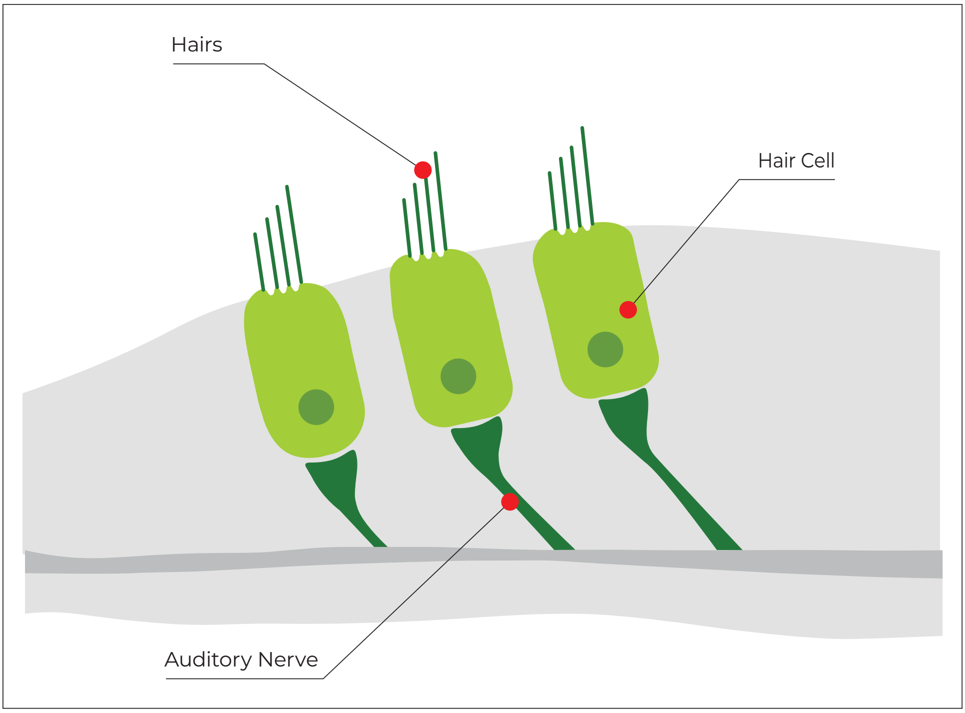


Figure 7.12 Magnification of the cochlea so that the hair cells responsible for the actual signal recording are visible.

Hair cells come in two different varieties: inner hair cells and outer hair cells.

You have about 15,500 of these cells—a much smaller number than for rods or cones. However, we won't examine the exact differences between the hair cells.

But a useful thing to remember is that hair cells absorb the vibrations from the liquid and bend as a result. Depending on the frequency and sound pressure, the bending of the hair cells is stronger or weaker. When a certain bending is reached, the hair cells "fire" a signal in the direction of the auditory nerve and communicate that a sound has been heard.

The cochlea is curved and has hair cells at different locations along the cochlear partition. Depending on the hair cells' locations, the hair cells fire at different frequencies. Thus, different sounds are heard because hair cells bend at other locations in the cochlea.

The Right Balance Starts in the Ear: Vestibular Perception

In the ear, another significant perception arises for the human body: *vestibular perception*. The vestibular apparatus (or the *organ of equilibrium*) consists of three semicircular canals, each of which contains sensory cells, as well as two otolith organs (*utricle* and *saccule*). While the canals are responsible for depth perception, speed perception is generated by the otolith organs.

The three canals are each equipped with tiny sensory hairs and meet at right angles. They are partially filled with lymph fluid. If the head moves, the fluid moves over the hairs. The resulting stimulus is passed on via the canals in the ear and thus reaches the brain. In this way, every movement of the head is perceived and processed.

For UI design, vestibular perception is often secondary to vision and hearing. Usually, you cannot influence what the users perceive via the vestibular organ with your solution. However, the situation is different if you design a solution that physically moves users. In this case, feedback from the organ of equilibrium can be influenced by your solution. An example of such a product would be a car that attracts the user's attention with short braking (braking jerk) in the event of an impending collision.

However, we won't deal further with vestibular perception in this book because it cannot or will not be used to communicate directly with users except in a few cases. However, it still can contribute to the mix of sensory impressions that affect the users of your solution.

Hearing and Hearing Impairments

Earlier in the text, we presented the optimal range for hearing (2,000 to 5,000 hertz). Human speech falls in this range, as does most music takes place. However, hearing outside this frequency range is possible. In general, our ears can perceive frequencies between about 20 and 16,000 hertz, where loudness, which is physically the *sound pressure level* measured in decibels (dB), plays a significant role. A change in loudness of 10 dB corresponds to a doubling of the volume.

In many cases, however, the range of audible frequencies reduces with age—especially in men. However, unlike presbyopia, this change is not due to a change in physiology but to preventable causes such as an environment that is too noisy. Damages to hair cells from noise mean they can no longer transmit stimuli to the auditory nerve. In the beginning, the hearing ability is reduced only slightly, which can be compensated for by increasing the volume. However, severe damage can lead to complete hearing loss.

Put simply, you can imagine that too loud music or loud noises in general “blow” the hair cells so that they buckle and can no longer straighten up and vibrate. The damage to the hair cells is irreversible, initially making distinguishing speech from background noise harder. When was the last time you attended a rock concert without hearing protection? Even if it sounds stuffy: don't do it. Your hair cells will thank you.

Men, in particular, are affected by hearing loss. It is assumed that this loss is mainly due to *environmental conditions*. Men increasingly work in environments that are (too) loud for the human ear and thus actively contribute to the damage. For example, noisy work environments often prevail at construction sites. To protect you (and us) from damage caused by loud noises, Germany passed, for example, the Workplace Ordinance or the Noise and Vibration Workplace Protection Ordinance, which, among other things, defines the permissible noise pollution at the workplace and sets limits. In the US, similar standards that must be followed by employers are enforced by Occupational Safety and Health Administration (OSHA).

For example, in Germany, the limit for ambient noise levels for an 8-hour working day is 85 dB, with noise peaks not exceeding 137 dB. In the US, OSHA requires employers to implement a hearing conservation program when noise exposure is at or above 85 dB, averaging over 8 working hours (OSHA, n.d.).

To help you develop a sense of loudness outside the workplace, Figure 7.13 shows how loud different sounds are. In addition, you can also use the picture to develop a feeling for sounds that are too loud.

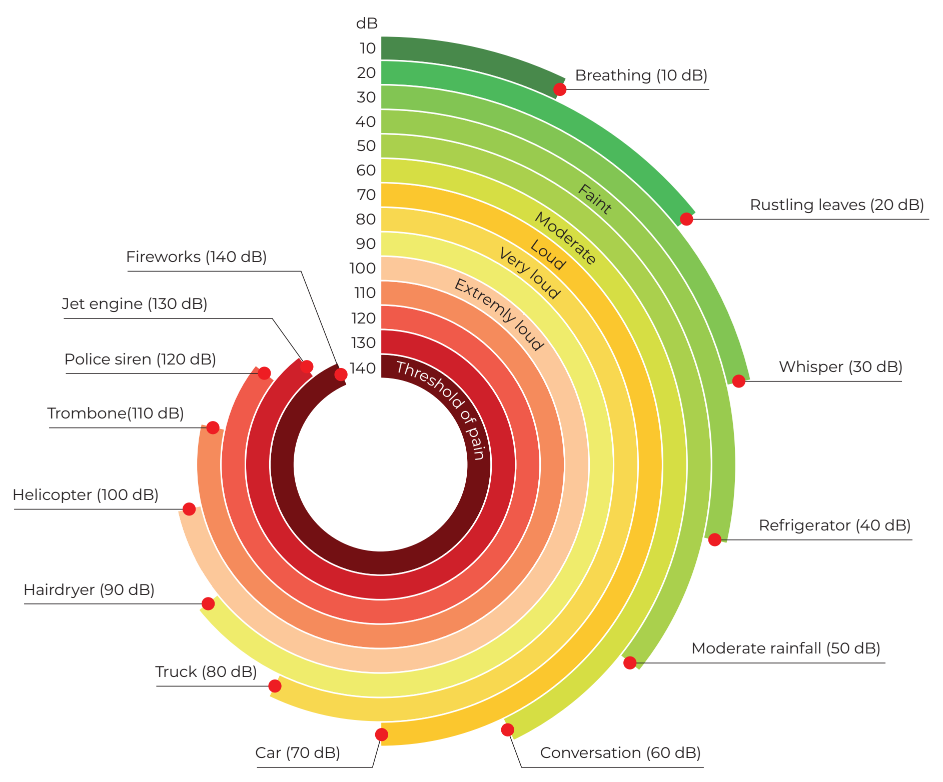


Figure 7.13 Different sounds and their typical loudness in dB. (Source: Modified from trgrowth/Shutterstock)

Key Takeaways for User Interface Design

So far, we've talked a lot about hearing. In this section, let's transform these characteristics of human hearing into design recommendations.

When you design sounds, make sure they are in an audible frequency range—preferably between 2,000 and 5,000 hertz—and they are at an appropriate volume to avoid harm. If your target group mainly includes older people, shifting the frequencies to lower frequency ranges, and thus, lower tones make sense, as does allowing users to choose the volume themselves.

As we mentioned earlier, a particular feature of acoustic perception is that auditory perception also works when you're directing your attention to something else. Conveying information visually will only reach your user if they look at it. Visual perception, therefore, depends on where the user is looking at the moment (including peripheral vision). Acoustic perception, on the other hand, has the advantage that you can also hear sounds behind you or in the next room, even if you're engrossed in a conversation.

Therein lies the great added value of acoustic information: You can use sounds and speech to attract attention! In some situations, sounds are helpful and desirable, for example, warning sounds and alarms. A smoke detector is one such example. In this case, a very loud sound is deliberately created to attract attention. Your attention shifts even if you're not in the same room and other sounds are superimposed. A smoke detector that warns visually only would not work as well. The same applies to an alarm clock that only shows the text "Wake up!" on its display.

In other situations, acoustic cues are perceived as very annoying. Haven't you ever been sitting in a subway, surfing on your smartphone, when a video comes out of a web page you open, blasting you and all the bystanders? In these typical situations, most people can do without sound perfectly well.

Sounds can therefore trigger a strong orientation reaction in people ("Where is that sound coming from?"). Use this knowledge to direct your users' attention in critical situations. However, the strength of acoustic signals is also a weakness: Other people cannot tune out these sounds. If you sit on public transport and talk loudly on the phone, others may not want to eavesdrop on you, but unlike your eyes, you can't easily "close" your ears. Thus, if you have sensitive information, for example, or other non-users are within earshot, you should primarily dispense with acoustic information or let the user start the audio.

However, you can use location independent of acoustic information even further. For example, auditory information is an excellent way to guide users if they cannot look at a display. Especially in the medical field, acoustic signals are often used. Examples of simple signals for guidance are alarms on patient monitors, as shown in Figure 7.14. In this case, a standardized alarm tone communicates urgency even without eye contact. Audible alerts are also often used during surgery. For example, surgeons more often have a handpiece (for example, a drill) with which to operate, which they operate via a foot switch (for example, the speed of the drill). But since the doctors must keep their eyes on the patient, an audible signal indicates whether the switch was pressed.

The idea of acoustic guidance is taken even further with defibrillators. In this case, the user is guided through interaction with the device solely through voice messages.

You're probably also familiar with a range of acoustic information that comes into play in more everyday environments. Think, for example, of the last time you stood at a train station and heard an announcement that your train was leaving today, not at track 3, but at track 18. Again, this information is communicated acoustically because the change on the display boards alone may not be detected in time by everyone. Similarly, the bell at the end of recess also clearly tells everyone the school day continues.

As you can see, on the one hand, you can use acoustic information to attract attention. On the other hand, you can also use it to provide direct guidance. Acoustic signals are an excellent complement to visual signals. But again, don't overdo it. People are indeed exceptionally good at perceiving various voice messages in detail and quickly. However, verbal language itself can make information take longer to convey in a meaningful way.

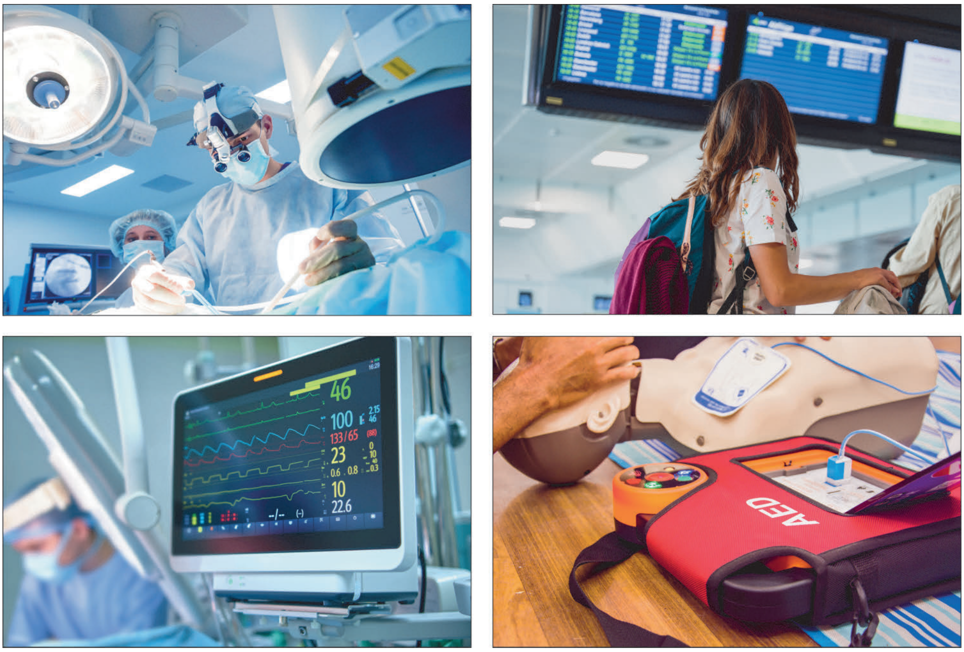


Figure 7.14 Examples of acoustic signals in medical and public areas: acoustic assistance for a surgeon (top left), announcement at a train station (top right), alarm tone at a patient monitor (bottom left), and acoustic guidance at a defibrillator (bottom right).

If you only use *warning tones*, then you theoretically have an extensive frequency range that you can use. But again, keep in mind *relative perceptual capability*, not *absolute perceptual capability*. Humans are indeed capable of hearing from 20 to 16,000 hertz. But that doesn't mean you have 15,980 warning tones at your disposal. When using warning tones, limit them to a few distinctly different ones and check whether standards have already been established.

In today's interactive media design, the eyes and the ears are the most important sensory organs. But the other organs also contribute. Therefore, let's take another look at the other sensory organs, always with direct reference to their usefulness for design.

7.2.3 The Skin

A vital organ of the human body, the skin serves as a barrier to the outside world and protects the insides of your body from dirt, bacteria, and other threats. In addition to its protective function, however, the skin is a huge window through which we perceive information about our surroundings. Unlike the eye or the ear, our skin only functions in close contact. So, we cannot sense over long distances. Nevertheless, much of the information we absorb is highly relevant—some information, such as pain perception, even ensures survival. Therefore, let's first look at the structure of the skin before we look at the skin's perceptive capabilities.

The Components of the Skin

Similar to the ear, the skin is divided into different areas. The skin comprises the epidermis, dermis, and subcutis, as shown in Figure 7.15.

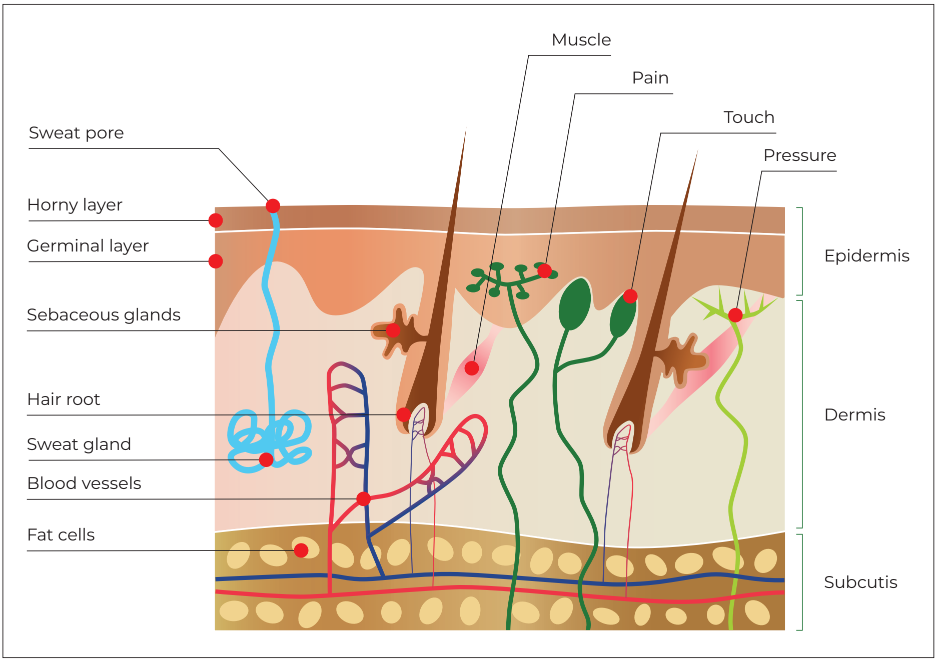


Figure 7.15 Skin structure with the three layers of the epidermis, dermis, and subcutis. (Source: Modified from https://de.wikipedia.org/wiki/Haut#/media/Datei:Schemazeichnung_haut.svg, Attribution: ShareAlike 3.0 Unported)

It may sound gross, but the *epidermis* consists of dead skin cells on the surface and forms the layer of the skin that we can see from the outside. Below are various layers needed for the formation of new skin. The different sensory cells in the *dermis* respond to stimuli from the outside. We'll take a closer look at this skin layer and its sensory cells in a moment since the dermis is particularly interesting to us as usability and UX designers. The *subcutis* is where the conjunctiva and fatty tissue are located. The subcutis thus serves as protection against cold but is only of limited importance for designing UIs.

In the dermis, different *sensory cells* process stimuli from the outside, as shown in Figure 7.16. The most common sensory cells are called *nociceptors*, which react to pain. These cells are distributed in different densities at different places on and in the body, with the highest density in the skin. Various pain receptors respond to mechanical action (pressure, stinging, etc.); thermal action (above 113°F or cold); and chemical action.

The nociceptors are considered *mechanoreceptors*, but they are not the only mechanoreceptors in the dermis. In addition to the sensory cells for pain, other sensory cells for processing lighter and less dangerous mechanical stimuli. Four different types of mechanoreceptors include Merkel cells, Meissner corpuscles, Ruffini corpuscles, and Pacinian corpuscles.

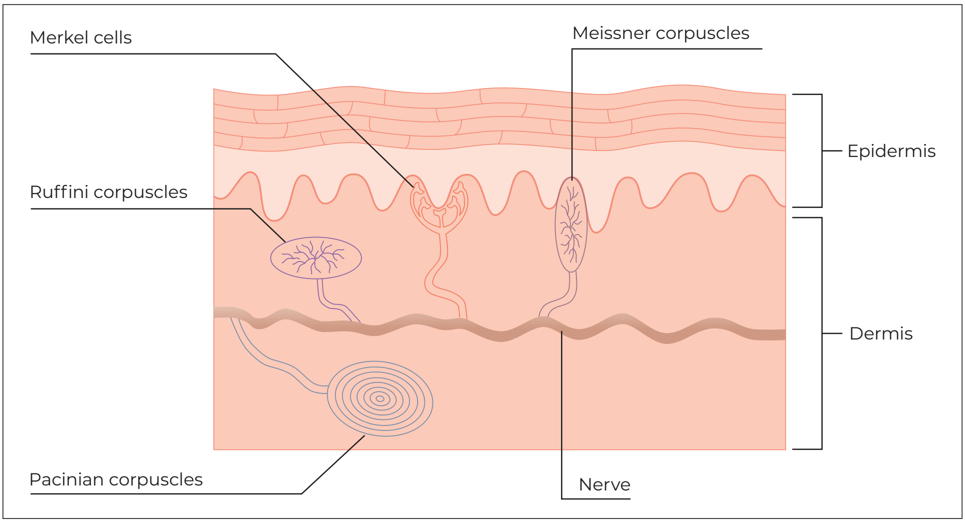


Figure 7.16 The different cell types and their perception. (Source: Modified from https://openstax.org/books/biology/pages/36-2-somatosensation#fig-ch36_02_03, Creative Commons Attribution 4.0 International)

Merkel cells are responsible for perceiving fine details, whereas *Meissner corpuscles* control your grasping force. The *Ruffini corpuscles* react to stretching of the skin, and the *Pacinian corpuscles* respond to vibrations.

A special feature of tactile perception is the great distance to the brain. Unlike the eyes, which are directly connected to the brain via the optic nerve, and the ears, which are connected via the auditory nerve, the skin is distributed over the entire body. This distributed nature results in two consequences:

1. The stimulus must be conducted through the body to the brain; the spinal cord serves this purpose.
2. The stimulus must be assigned in the brain; so, it must be determined which stimulus comes from where.

This stimulus transmission usually works very well and very quickly. However, accidents, for example, can lead to disturbances manifesting in total or partial paralysis.

The assignment of the stimuli to the causative body area also works well. However, Figure 7.17 shows a weighting of the skin regions in the brain. Certain areas, such as the fingertips or lips, are overrepresented and thus much more sensitive than, for example, the shin. And you know what we found fascinating? Your brain is adaptable in the process. If you do something particularly frequently (for example, playing a stringed instrument), the corresponding brain areas become even more pronounced.

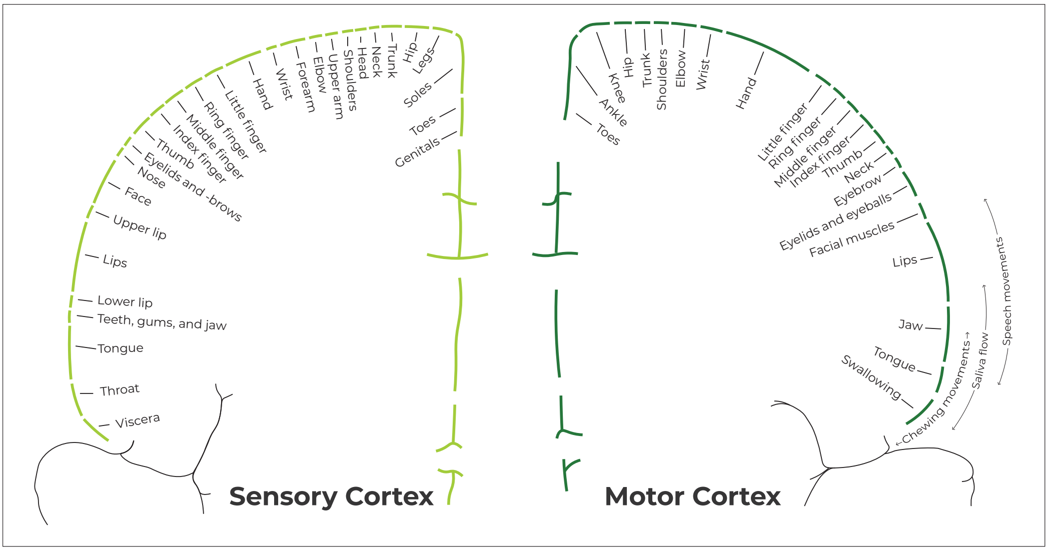

Viscera
Throat
Tongue
Teeth, gums, and jaw
Lower lip
Lips
Upper lip
Face
Nose
Eyelids and -brows
Thumb
Index finger
Middle finger
Ring finger
Little finger
Hand
Wrist
Forearm
Elbow
Upper arm
Shoulders
Head
Neck
Trunk
Hip
Legs
Soles
Toes
Genitals
Sensory Cortex
Toes
Ankle
Knee
Hip
Trunk
Shoulders
Elbow
Wrist
Hand
Little finger
Ring finger
Middle finger
Index finger
Thumb
Neck
Eyebrow
Eyelids and eyeballs
Facial muscles
Lips
Jaw
Tongue
Swallowing
←Chewing movements→
Saliva flow
Speech movements
Motor Cortex

Figure 7.17 A homunculus according to the Jasper-Penfield scheme of Penfield & Rasmussen, 1950. A full PDF of this image is available for download at www.rheinwerk-computing.com/5778 under the product supplements.

Feeling and Limits of Tactile Perception

High sensitivity also makes our skin an excellent medium for communication. *Braille* was developed as early as 1825 to enable blind and visually impaired people to read using their sense of touch, as shown in Figure 7.18.

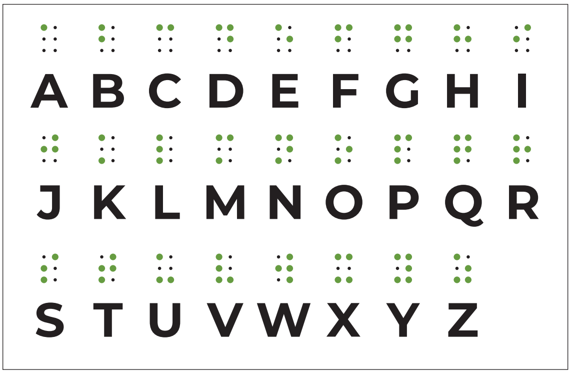


Figure 7.18 Braille alphabet.

In Braille, the individual letters of the alphabet are formed by a combination of dots, which can be in six different positions. The result is another letter depending on which dot is raised and which is not.

Crucial for the functioning of Braille is what's called the *2-point threshold*. The *2-point threshold* is the distance above which a person can recognize two individual stimuli (e.g., pen tips) as separate stimuli. Below the 2-point threshold, the pen tips are perceived as one tip. This 2-point threshold varies depending on the body region, as shown in Figure 7.19.

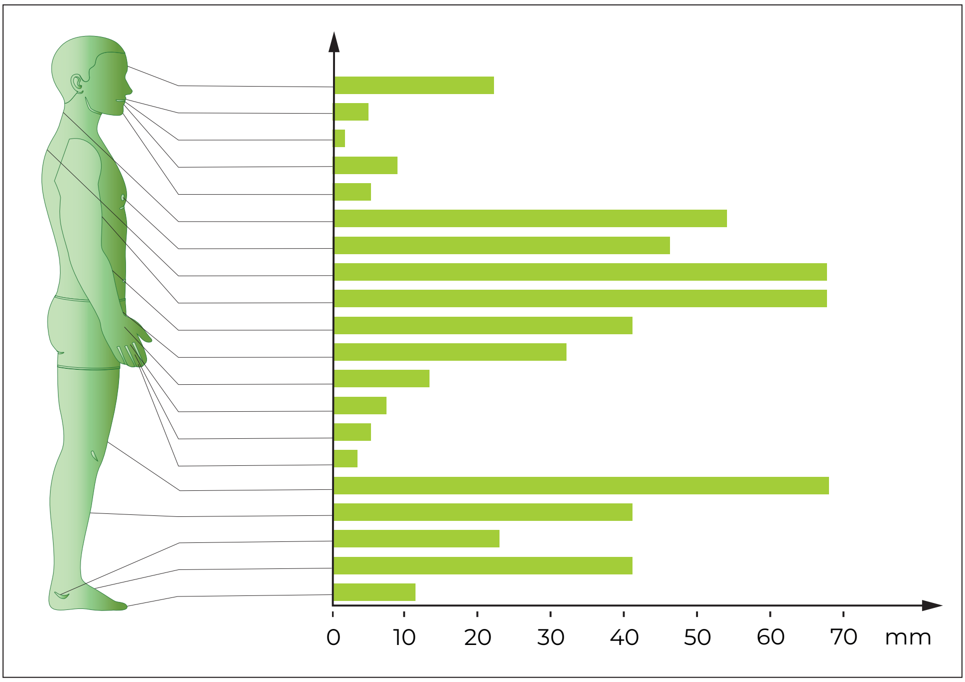


Figure 7.19 2-point threshold for different body regions. The smaller the value, the more sensitive the respective area. (Source: Modified from Weber, 1835)

The 2-point threshold is lowest in the area of the tongue and fingertips. However, if you tried to read Braille with your toes, the individual dots must be about 0.4 inches apart. On the lower thighs, you would need dots about 2.6 inches apart. Imagine a whole text written in this size!

Key Takeaways for User Interface Design

Consider the *2-point threshold* if you're a usability and UX designer working on physical products. But also, if you're planning new input devices, you should think about how well certain areas of the body are suited to a task. For example, foot controls often come into play in the medical field. Compared to handhelds, these input devices are significantly larger and require much less accuracy. This scenario is partly due to the lower level of attention given to the foot compared to the handheld part being worked nearby but also due to the significantly lower sensitivity and associated accuracy you can achieve with your feet.

In addition to pressure and stretch, an essential way of perceiving the skin is the perception of *vibration*. This ability makes it possible to transmit information via different vibration patterns. In the design of UIs, vibration is primarily used for silent and discreet data transmission. An example is a smartwatch or sports wristband, shown in Figure 7.20, to indicate the arrival of a new message or to wake up a sleeping person without forcing their partner to wake up.

Recently, the use cases for vibration have continued to increase. For example, navigation belts now use vibration to enable orientation and navigation without relying on eye contact with a display. Some smartwatches also offer the same functionalities.

Figure 7.20 Sports wristband with vibration function as an alarm clock.

These technologies are currently used whenever the primary information channel (the eyes) is not available—for example, due to a visual impairment; due to heavy smoke development (in safety technology, as shown in Figure 7.21); or because the user's gaze should not be taken off the road (for navigation).

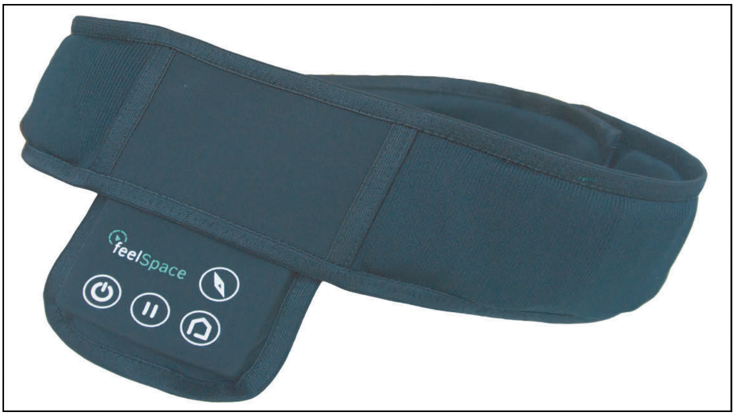


Figure 7.21 Navigation belt from feelSpace. (Source: www.feelspace.de)

However, a conceivable future trend is that usability and UX designers will use this potential more frequently. The first providers have already developed touch displays where various visual elements can be used (as with conventional screens), and tactile feedback can be defined simultaneously with pinpoint accuracy.

This accuracy allows you to display ten buttons on the same surface, each with a different feel. If the vibration is implemented well, pressing a digital button on a touch display suddenly feels like pushing an actual button. As a usability and UX designer, you can break out of the flat screen world, at least to a certain extent.

The use of vibrations and tactile feedback is also playing an increasingly important role in the games industry. Many manufacturers are currently working on simulating feedback via the skin in addition to a visual image of virtual reality. This feature is intended to significantly increase the feeling of being in the world—what's called *immersion*—compared to purely visual/audible feedback.

In this context, however, the limits of the skin also become apparent. As already described, our skin is an instrument for close perception. Thus, if you use the skin for your interaction, you will only ever succeed by stimulating the skin directly. In virtual reality, suits, vests, or gloves are used for this purpose, as shown in Figure 7.22.

But the skin also has advantages over our eyes. Like the ears, you can use the skin to transmit signals independent of your current attention. Your skin doesn't have to "look" anywhere to feel vibrations or pressure, which is what makes the skin so effective for us as a protective mechanism. Especially in situations where you need a design that warns users regardless of their attention, you can use this capability. For example, a few years ago, we worked on a project that used cold to warn construction workers of an approaching hazard.

Figure 7.22 Gloves for increasing immersion in virtual reality through direct stimulation.

If you're interested in haptic perception, we recommend the book *Human Haptic Perception—Basics and Applications* by Martin Grunwald.

As a brief and thought-provoking conclusion to this section, we would like to briefly point out that the nociceptors (reminder, the pain receptors) can theoretically also be considered by you in the interaction design. Don't believe it? At least in dealing with animals, it is common practice. An electric fence surrounding a pasture has a "UI" that makes it clear to the users (in this case, the animals) through pain perception that this is as far as they can go.

But also, in human interaction, some (few) applications rely on the sensation of pain as an interface. One example is social games where friends can be given mild electric shocks, as shown in Figure 7.23—either because they have been caught lying or react too slowly to a signal. And you thought you didn't need to know about pain receptors!

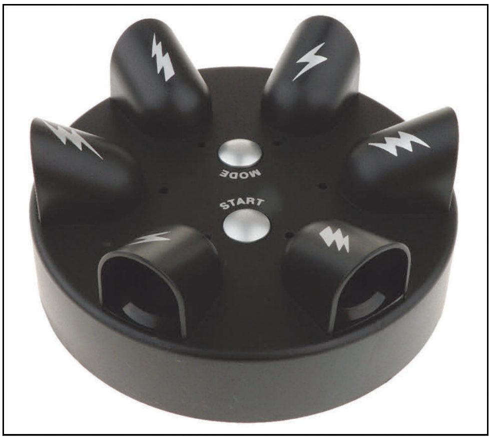


Figure 7.23 A parlor game that uses electric shocks to target pain receptors.

7.2.4 The Tongue

The tongue carries different taste buds and enables us to distinguish between various flavors. We mainly perceive food and understand its digestibility through our tongues. In this section, we'll again present its structure before exploring its perceptual capabilities and the benefits for you in usability and UX design.

The Components of the Tongue

The tongue, just like the nose, is often considered a gatekeeper. Unlike other sensory systems, our tongues primarily come into contact with substances we ingest through our mouth. So, our tongue and nose have the task of deciding, at lightning speed whether, what is in our mouth should reach our stomach.

Harmful substances have the "advantage" that they often taste and smell unpleasant. In contrast, healthy substances often taste and smell pleasant (or evolution has developed us in such a way that we perceive this). Whether unpleasant or pleasant, in both cases, the taste/smell triggers strong emotions that cause us to stop immediately or continue eating.

Approximately 10,000 *taste buds* are distributed on the tongue to make this possible. Each of these *taste buds* has 50 to 100 *taste cells*. Different *taste receptors* exist on these taste cells, each responding to other substances and triggering our typical taste perceptions. Depending on the location of the taste buds and taste cells on the tongue, the stimulus is then transmitted to the brain via a different nerve pathway.

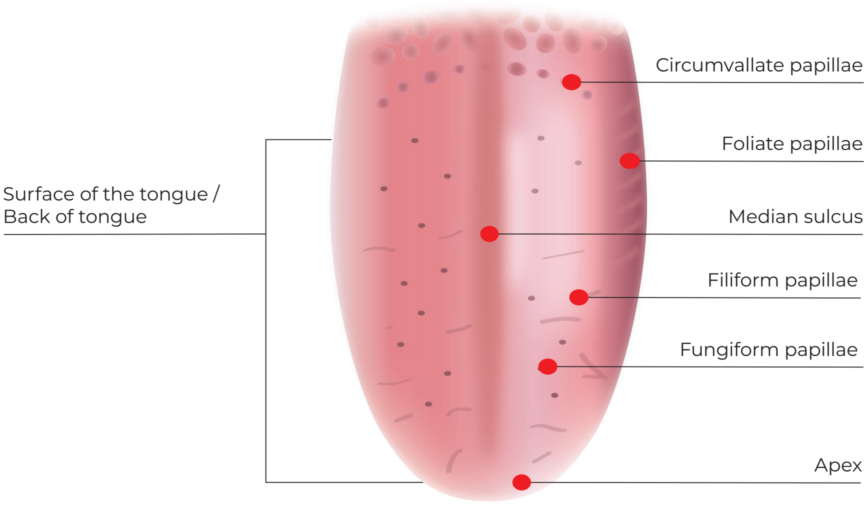


Figure 7.24 The tongue.

The tongue itself looks pretty rough and furrowed. This texture is due to what's called *lingual papillae*. Four different types of lingual papillae exist (Gesundheitsinformation 2021), as shown in Figure 7.24:

- Circumvallate papillae
- Foliate papillae
- Filiform papillae
- Fungiform papillae

Each of the papillae types contributes to the unique appearance of the tongue. Filiform papillae, for example, are distributed over the entire tongue and provide the rough look. Fungiform papillae look like tiny mushroom hats and are visible as an elevation on the tongue. Foliate papillae form the indentations at the back edge of the tongue, and circumvallate papillae sit like small, hilly craters at the back end of the tongue. The taste buds are located within the various papillae.

Taste Perception

Contrary to the long-held assumption that we can distinguish four different tastes, the current state of science is that five different flavors can be perceived via the tongue: bitter, salty, sweet, umami, and sour, as shown in Figure 7.25.

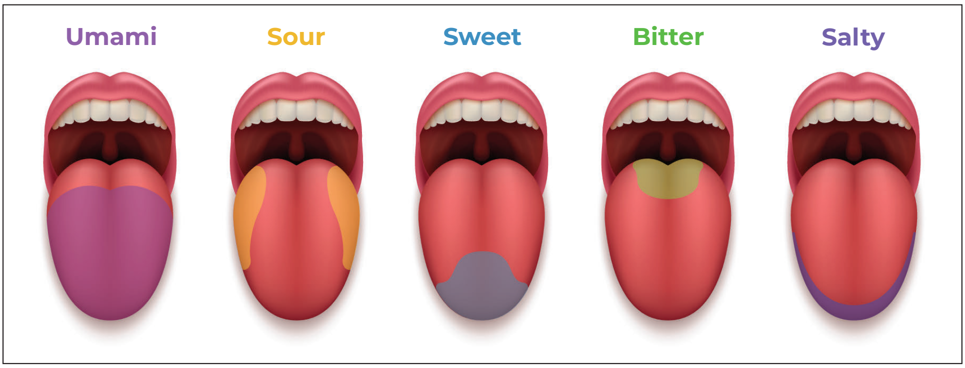


Figure 7.25 Taste zones of the tongue. (Source: Modified from MicroOne/Shutterstock)

For most taste receptors, perception of a particular taste predominates in certain areas. Sweet taste, for example, is located at the tip of the tongue, and bitter taste, at the back.

In this context, taste perception is less easily considered a single result than the sensory systems already presented. Unlike the eye, ear, or skin, the tongue and nose work together when we perceive the aroma of food. Researchers now predominantly speak of a gustatory "basic quality" when they mean sweet, sour, salty, bitter, or umami.

Key Takeaways for User Interface Design

At present, tasting plays a minor role in the design of UIs, although this lack is increasingly changing. For example, a television was presented for the first time that can be licked by the user ("Taste the TV".)

In this way, some manufacturers can ensure that users are more involved in the digital world via the tongue and experience a richer perception of their surroundings. The same research team is also developing this technology for forks, for example, to improve food taste.

If you plan to do something similar, please think carefully about the consequences. Simulating taste is a fantastic idea, and we believe a largely unoccupied area is open to all creative minds and technology fans here. However, we still doubt whether this makes sense for a device several people use in parallel or one after the other....

7.2.5 The Nose

The nose is our second sensory gatekeeper. It, too, is a warning system that alerts us humans at an early stage. Do you know the moment you open the refrigerator that something has perhaps been there a bit too long? Usually, we consciously smell food before putting it in our mouths. The same thing happens unconsciously, even with less noticeable food. In this way, sometimes, even before the food touches the tongue, we can determine that something is wrong with it, and the food intake process can be stopped.

Let's see what components allow us to detect spoiled food and what other abilities our noses have.

The Components of the Nose

The human sensory system for smelling is located in the nose and is called the *olfactory mucosa*. The olfactory mucosa is approximately 0.8 inch^2 in size and is located far up in the nasal cavity (Anatomie Mensch, 2021), as shown in Figure 7.26.

The substances we later smell are carried as individual molecules through the air into our nose, reaching the olfactory mucosa. The olfactory mucosa contains olfactory sensory cells.

The human olfactory mucosa has approximately 10,000,000 olfactory sensory cells of each receptor type. That sounds like an incredible number. Even compared to other sensory cells, such as the taste buds, that's a lot. However, following Goldstein and Cacciamani's (2021) classification, we humans are what are called *microsmatics*. We rely less on our sense of smell than other creatures. Opposite us are the *macrosmatics*. Maybe even a macrosmatic lives at your home: Dogs have one of the most sensitive noses. Compared to humans, dogs have not 10 million but 1 billion olfactory sensory cells for the different receptor types. Depending on which odor is involved, their nose is 300 to 10,000 times more sensitive than a human nose, as shown in Figure 7.27.

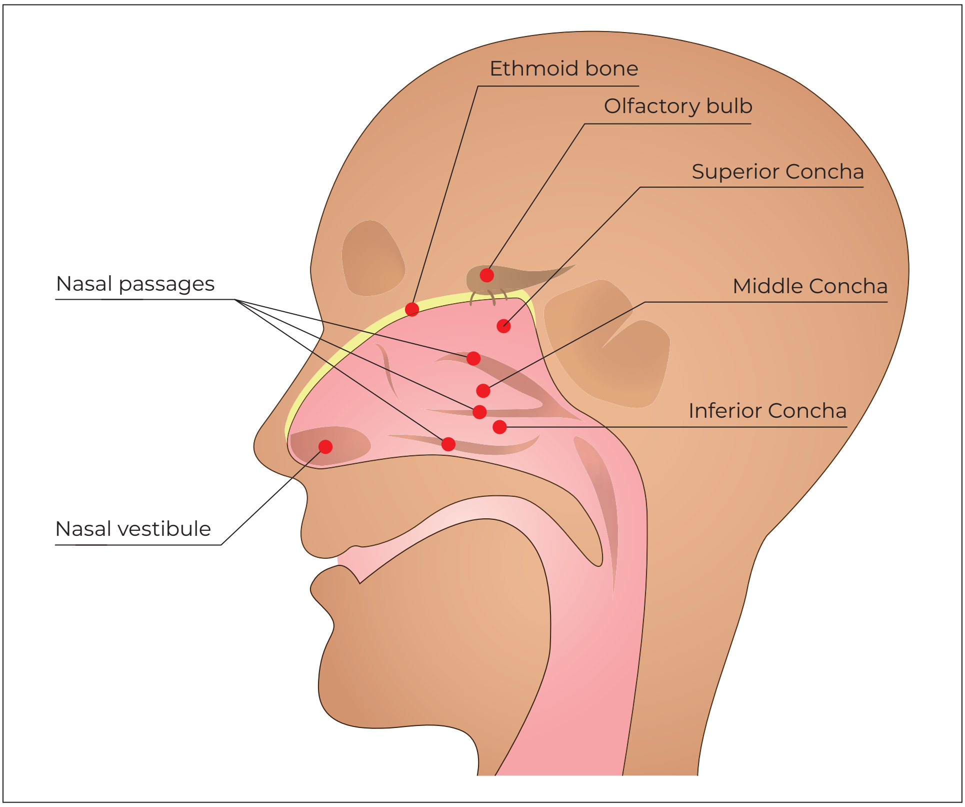

Ethmoid bone
Olfactory bulb
Superior Concha
Nasal passages
Middle Concha
Inferior Concha
Nasal vestibule

Figure 7.26 Components of the nose important for UI design.

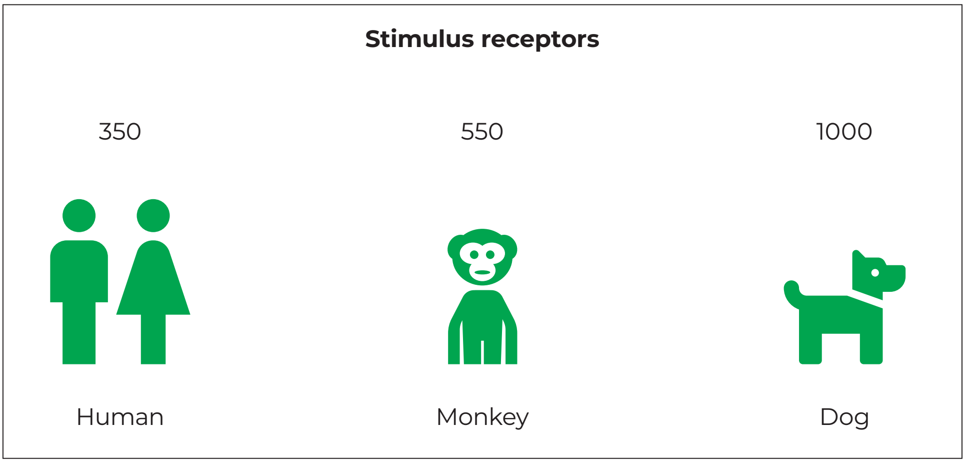


Figure 7.27 Comparison of the sensitivity of human and animal noses.

To perceive the different odors, olfactory sensory cells have different olfactory receptors, each responding to different substances. Thus, the structure of our olfactory perception is initially similar to the structure of other sensory perceptions. Interestingly, however, smelling is much more complex. While our eyes manage with four different receptors (rods and S/M/L cones), more than 350 different receptors are involved in olfaction. We are still microsmatics. In the mouse (you guessed it, a macrosmatic), there are about 1,000!

In macrosmatics, the sense of smell plays a crucial role in orientation in space, marking territory boundaries, or even mating. Less so for us humans. Although initial indications suggest that men can smell a woman's fertility, this hypothesis has not yet been proven on a large scale. Overall, Goldstein and Cacciamani (2021) concluded that smelling is more of a luxury function for us, one that significantly increases the quality of life but is only conditionally necessary for survival.

However, even in humans, the sheer quantity of different receptors means that odor perception is quite complex. Odors are usually composed of a response from different receptors. The result of the different receptor activations is called an *odor profile*.

The *olfactory bulb* is directly above the olfactory mucosa. Nodes called glomeruli traverse this structure. Each *glomerulus* is a node where all olfactory receptors of the same type converge. The glomeruli represent the last essential component of the olfactory sensory system.

Now that you have learned about all the parts, let's consider the possibilities they give us humans for our olfactory perception.

Olfactory Perception

The human sense of smell is less sensitive than some animals but still quite powerful. Depending on the substance to sniffed, even tiny amounts are enough. For example, less than 1 ppb (the number of molecules of a particular substance in a billion molecules of the carrier substance—in this case, usually air) is enough to detect the odorant *butyl mercaptan*, which is typically added to natural gas. With acetone (contained in some nail polish removers), on the other hand, perception doesn't occur until 15,000 ppb.

As you can see, our noses can also be susceptible. However, since many possible substances exist, it is difficult to state the nose's sensitivity clearly. The *detection threshold* is the point from which an odor can be perceived.

In addition to the detection threshold, smells have what are called a *recognition threshold*. From the detection threshold, we only establish that something smelly is there. From this point, you can classify the odor's quality, which includes determinations such as "flowery" or "minty." A factor of about three exists between the detection threshold and recognition threshold; in other words, three times as many ppb in the air is required to recognize the odor as opposed to detect it. However, the thresholds depend on the substance, so giving exact specifications isn't easy.

What is feasible, however, is a statement about relative sensitivity. Studies have compared the different percentages at which people can sniff out different substance concentrations. The result is that the concentration must differ by at least 11% to be perceived by people as different in strength.

Also of interest on perceptual abilities for odor, humans are believed to distinguish up to 100,000 different odors, but many people find it challenging to identify an odor accurately.

In other words, you can likely say, "No, that smells different," but it's doubtful you can say, "Oh, that's a tulip!" Interestingly, this problem is primarily a naming problem, and once it is known what the smell is, it is perceived that way.

Key Takeaways for User Interface Design

The sense of smell is challenging to grasp for the design of products. Perhaps for this very reason, smell still plays a subordinate role in the design of interactive media. But in this area, too, we expect a change in the next few years in which smells will also be used to increase immersion in digital worlds. Unlike tasting, smelling can be used over greater distances, making it easier to use. Let's face it, if a lickable TV has already been invented, olfactory TV can't be far behind.

But seriously, approaches to product design based on smell already exist. *Air Up* was the first to offer the concept of giving water a "taste" using scents. The promise behind it: You can have sugar- and calorie-free water and still feel like you're enjoying a delicious beverage. The promise pulls. In just a few years, the company has grown from just 5 founders to more than 300 people and gained over 2 million customers (figures from 2023). So, smelling can contribute to success even today.

But this scenario is also where the peculiarity of the smell becomes apparent. How do you drink water but taste something different through the smell? The senses of taste and smell are extremely closely linked in humans. Much of our taste perception comes from the interaction of odor and taste. If either is missing, the experience quickly becomes bland. You're probably familiar with this effect in your own life: Think back to the last time you had a nasty cold. Your nose was closed entirely, and you could hardly breathe. No food tasted good then, either.

7.2.6 Conclusion

Since ancient times, the sense of sight has been assumed to be the most important of all the senses and even that up to 70% of all sensory impressions reach people by visual means. However, recent studies suggest that the question of which sense is dominant in people is strongly linked to culture and personal origin.

In one study, Asifa Majid and colleagues (2018) showed that, depending on the culture, other sensory impressions could be described much more clearly in language and are thus more in focus for that culture. In this context, our Western culture is predominantly characterized by the eyes, whereas in the Middle East, a dominance of the sense of taste was found. Across the cultures studied, quite different primary senses were used in each case, with only the sense of smell being found to be quite elusive for most cultures. The only exception to this trend seems to be cultures that still intensely focus on hunting and gathering.

Regardless of culture, Figure 7.28 shows the different sensory systems we all have.

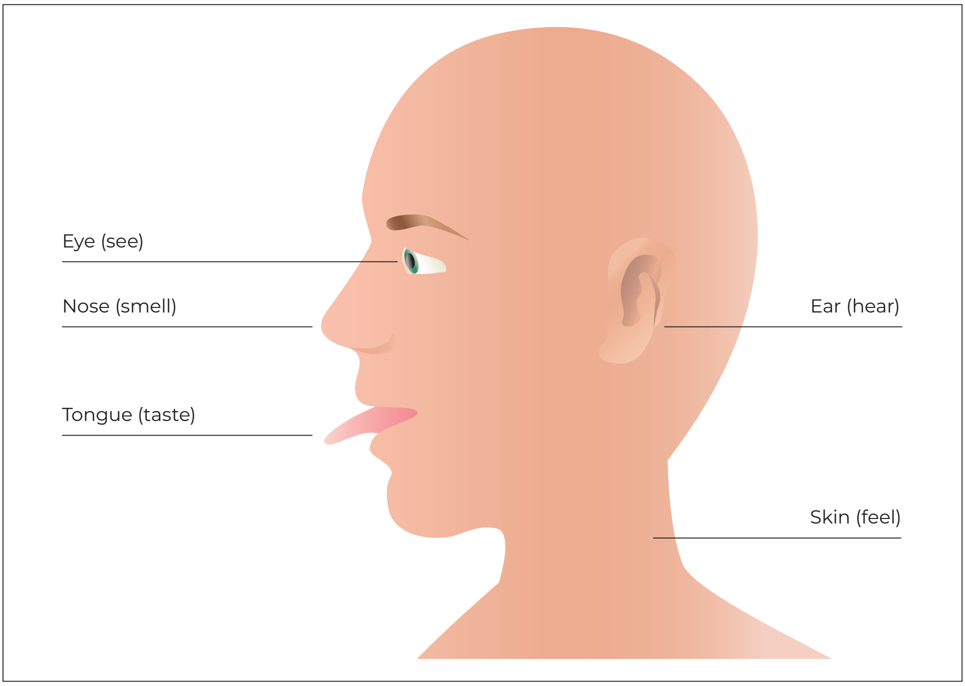


Figure 7.28 Sensors of the human body.

However, our sensory system is only a part of our perception. A large part of perception occurs not in the sensory organs but in our brains. Therefore, the following section will examine how our brain processes information.

7.3 Human Information Processing

Up to this point in this chapter, we have dealt—more or less in detail—with humans' most important sensory systems. We've tried to be as descriptive as possible to capture all there is to human sensory technology and what it can do. In each section, we tried to relate it to you and your work as a usability and UX designer.

Nevertheless, knowing the human sensory system is only a first step to understanding human perception and information processing in general. In this chapter, we would like to illuminate the "second" side, namely, what happens when the sensory system has reported, "Here's something!"

As you can imagine, human information processing is hugely complex. The book *Human Information Processing* (Lindsay & Norman, 1977) alone devotes more than 770 pages to the topic. For this book, however, our focus is on usability and UX design. We'll provide you with essential basics but steamroll through the whole thing and present only a single, simplified model. But don't worry: This simple model works great for our purposes. In addition, our model has the advantage of being quite easy to remember because of its simplicity and, thus, quite easy to use in practice. We are talking about the *simplified stimulus-response model*, as shown in Figure 7.29, which is based on Luczak (1975).

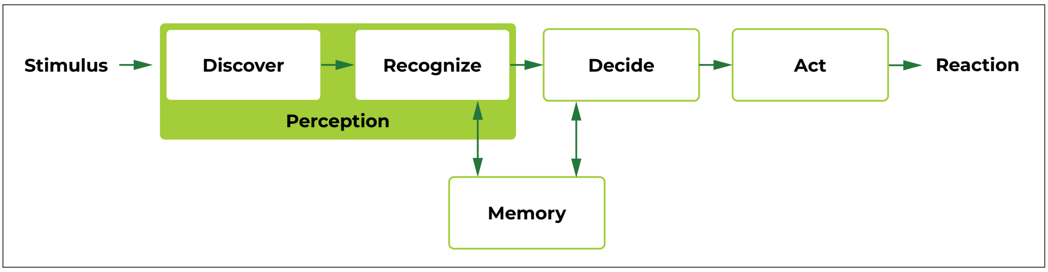


Figure 7.29 The simplified stimulus-response model according to Luczak, 1975.

The model itself, while pretty old, is still relatively young compared to human history. Basic human information processing does not evolve that quickly. Let's now look at this model together.

7.3.1 Basics of the Model

The simplified stimulus-response model of human information processing is based on the idea that information processing takes place in individual sequential stages. The next step can start only when one stage has been completed.

For our model, this means fast forward (don't worry, we'll go through everything in detail later): First, a stimulus must be present. This must then be discovered before it can be recognized. Following this, a decision takes place, which leads to an action. The action is then the reaction that can be observed in the external world. Memory also influences recognition and decision-making.

In addition to the sequential process, the second important idea behind the model is that human information processing takes time. Only when the required time has been invested in one stage is it possible to advance to the next. Thus, the more stages are passed through, the more time must be invested. In fact, this model was proven with more or less elaborate experiments, and the times for the individual stages were calculated using simple examples.

You don't need to remember how long an individual step takes. Key takeaways include the stages of information processing and what happens in the individual stages.

Tip

We recommend that you read this section particularly carefully. You'll come across this model again in our discussion of analyzing errors with root cause analysis (see Chapter 12, Section 12.22). At least in the development of medical devices, this model is firmly anchored.

A note in advance: As the name suggests, the model simplifies human information processing. Therefore, not all observable processing steps can be explained entirely by this model. However, this comprehensiveness is not our claim; we want to introduce the model for practical use. Let's go!

7.3.2 Stimulus

At the beginning of information processing, a *stimulus* always exists. Recall from Section 7.2 how the stimuli that humans can process are incredibly diverse. The stimulus that triggers information processing can be light, a sound, a touch, a change in temperature, or a chemical substance, as in the case of smelling or tasting.

In most situations, we are pelted by many stimuli, but only a few actually reach us. The filter between the stimulus and us is, in many cases, *attention*. With our attention, we can—more or less—consciously control which stimuli reach us and which do not. As described earlier in Section 7.2, this control is not available for every stimulus. For example, you cannot ignore sudden pain.

However, we do not consider attention itself in this model, even though there would, of course, be much more to say about it.

What you should take away from this section is that, without stimulus, there is no information processing. In principle, the stimulus could also be an internal one. For this book, however, we'll focus on external stimuli because, as a usability and UX designer, you primarily work “outside” humans.

7.3.3 Discover

The first step in the human being is the *discovery of a stimulus* Discovery is simply a matter of filtering the stimulus out of the abundance of existing stimuli. Discovery forms the first step of *perception*, as shown in Figure 7.29.

For discovery to work, the stimulus must fit the human senses, environment, and current state.

Imagine building a product that tells people there's a problem using a warning light. So far, so good. But now, suppose that, for availability reasons, one decided to use a warning lamp with pure UV light, which is outside the visible spectrum of the human eye. So, there would be no chance for the warning to be detected, as shown in Figure 7.30.

Figure 7.30 The UV lamp tries to warn you intensively... or does it?

But even less clear-cut cases can lead to a stimulus not being detected. Imagine you're driving on the highway and would like to pass the car ahead of you. Pulling to the left, you hear a loud honking behind you. Apparently, another vehicle was next to you, but due to the blind spot, you could not detect the other car. Thus, you might not discover a stimulus because it was simply undetectable from your position, as shown in Figure 7.31.

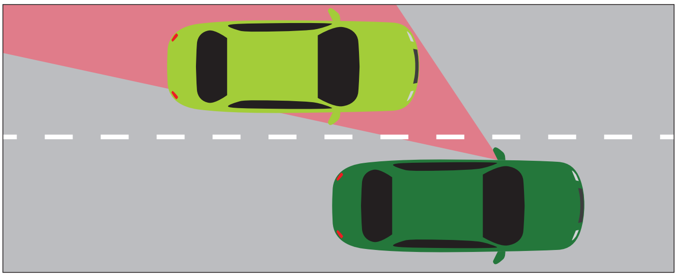


Figure 7.31 A vehicle in your blind spot. The stimulus is there but not always discoverable.

Let's consider another example: Imagine standing outside in the sunshine and wanting to read something on your smartphone. Due to the intense sunlight, you can't see any writing on the display but simply a bright surface, as shown in Figure 7.32—another case where you would not detect the stimuli.

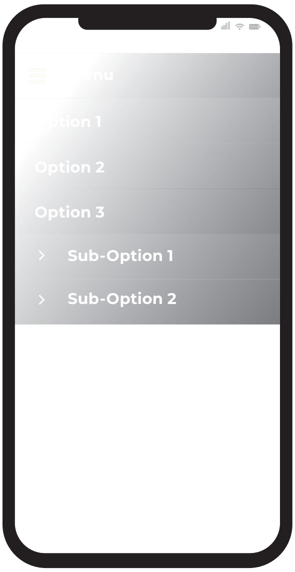


Figure 7.32 Smartphone display in bright sunshine: a stimulus barely detectable to the human eye.

Therefore, whether you discover a stimulus depends on whether it suits you and your perceptual possibilities. We have also considered that these can be partially limited (for example, due to color vision deficiencies) or changes in the course of life (for example, due to presbyopia).

On the other hand, discovery also depends on whether the environment allows detection on your position relative to the stimulus. For example, placing visual displays, screens, and so on behind a person only works in rare cases.

As mentioned earlier, one basic assumption of the simplified stimulus-response model is that the individual stages require processing time, which can be described easily with this example. If a display is placed behind a person, the visual stimulus emitted from the display will eventually be discovered—but the person may not turn around for hours. Discovery is probably much faster if the display is in front of the person directly in the field of view. Strictly speaking, the human is not stuck in the discovery stage but passes through all stages at various times. In the practice of usability and UX design, however, it is often about discovering a particular event. We neglect this point for the explanation and continue to speak of a "slow discovery," for example.

How quickly you detect a stimulus depends, on the one hand, on the stimulus strength and "placement." In addition, however, the discovery also depends on the person and here we are again on the topic of *attention*. For example, if you're engrossed in this book (which we strongly assume), you'll tune out some other stimuli in your environment. Also, the probability of overlooking a stimulus or discovering it later would increase significantly if you were tired or drunk.

The model excludes personal conditions, which you must also consider, however. For example, if your product is used in a fire station with on-call duty, then the users are probably more tired than average or first must be woken up before the stimulus that you set with your system, product, or service is discovered.

Sometimes, however, quick discovery alone is the goal of the design. For example, for a high-visibility vest, the primary goal is rapid discovery.

To promote discovery, a wide variety of options in the field of usability and UX design are available: You can work via color, size, shape, sounds, and blinking. The more unusual a stimulus is in its environment, the more likely it is to be discovered. But what happens to a stimulus once it has been discovered?

7.3.4 Recognize

Imagine that you have managed to turn your smartphone display to a position where you can discover the content on the display, despite the bright sunlight. In the discovery step, your brain tells you, "Watch out, there's something there." *Recognizing* now involves whether you can derive meaning from what you see, hear, smell, taste, or feel.

Imagine that, on the screen, the characters shown in Figure 7.33 are displayed.

Τηισ ισ α τεξτ ιν Ωινδοωσ φοντ

Figure 7.33 You have discovered these characters. But can you recognize them?

You might well discover the characters (you know they are there) but not be able to recognize them.

In this example, the characters encode the sentence "This is a text in Windows font"—written with the "Symbol" font from Microsoft Word. Recognition, then, concerns whether you understand what you've discovered.

The more you learn personally, the easier it is for you to discover certain stimuli and recognize them. In the diagram of the simplified stimulus-response model of information processing, this relationship is covered by the arrows pointing in the direction of *memory*, as shown in Figure 7.29.

Your memory has two functions in this context: On the one hand, you might remember that you have seen something like this before. On the other hand, you may have already stored a meaning to it. For example, a small child may remember seeing an elephant before. However, more than one elephant is probably required before they can remember that the large gray animal is called an "elephant."

Transferred to the Symbol font characters, recognition might involve the following steps:

- You see the signs for the first time and do not recognize them (discover but do not recognize).
- You see the characters for the second time and now know they are characters in Symbol (discover and partially recognize).
- You see the characters repeatedly, have memorized them, and can now read the text in Symbol (discover and fully recognize).

Again, the reference to time can be derived from this component. It is directly apparent that the three mentioned points are based on processing times of different lengths. Suppose you can read Symbol; recognition is fast. If, on the other hand, you first must understand which signs are being displayed and research their meaning, hours could pass. (In this case, the complete information processing process would run through several times and not stop at one point, which we neglect for reasons of complexity.)

Of course, perception in this form does not only work with characters. Imagine you're driving toward a traffic light. When you discover the traffic light as a stimulus, you first see some structure in the direction of the stimulus on which a green light is lit. But if you have now recognized that the structure is a traffic light, you also recognize that the light is green and know that this color means you're allowed to drive.

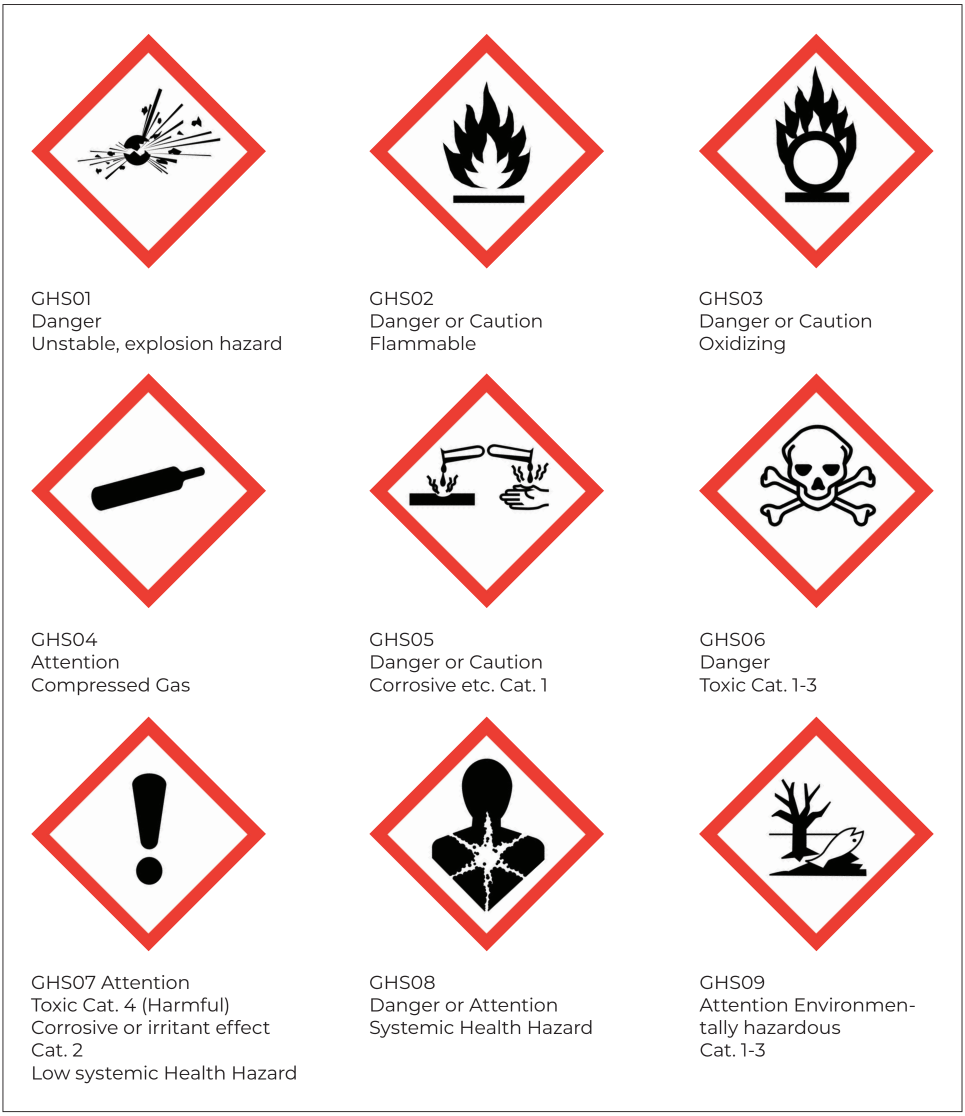
To know the meaning of the green traffic light, you must have learned it. From our experience, it is easiest to understand that learning the meaning is the difference between discovering and recognizing, using toddlers as an example. If you observe toddlers, you'll hear them enthusiastically call out repeatedly: "Mommy/Daddy, look!" They usually point to something they have just discovered. Shortly after that, they often ask: "What is that doing?" So, discovery works excellently, but recognition is not yet complete because still few contact points have occurred.

In addition to recognizing increasingly diverse stimuli, information processing functions more quickly with increasing familiarity. For example, situations can be assessed as a whole and not just as individual components. Clearly, the more that has been learned, the better the assessment of the situation works and the faster you are.

This situation is important for your design because, as a usability and UX designer, you must know that "having seen" does not mean "having understood." Especially if you work with icons in your design, you should remember this fact regularly! A wide gap in recognition exists between experienced and under-experienced users with a specific product or field.

Operating in a regulated environment, you can optimize recognition using *standard symbols*. Even if you're free in the design, you should fall back on symbols already established from a usability point of view. If you still use your own or new symbols, you should test them for comprehensibility and speed of interaction.

Even when using plain language, you may naturally exclude people who (don't) speak this particular language by your design. So please take away: Discovering a stimulus doesn't mean understanding it. Test yourself with the symbols shown in Figure 7.34. You can probably discover all the signs, but can you also recognize their meaning in each case?



GHS01
Danger
Unstable, explosion hazard

GHS02
Danger or Caution
Flammable

GHS03
Danger or Caution
Oxidizing

GHS04
Attention
Compressed Gas

GHS05
Danger or Caution
Corrosive etc. Cat. 1

GHS06
Danger
Toxic Cat. 1-3

GHS07 Attention
Toxic Cat. 4 (Harmful)
Corrosive or irritant effect
Cat. 2
Low systemic Health Hazard

GHS08
Danger or Attention
Systemic Health Hazard

GHS09
Attention Environmentally hazardous
Cat. 1-3

Figure 7.34 Various danger signs, all of which you can probably discover. Can you recognize all of them? (Source: Modified from https://www.productip.com/kb/productipedia/compliance-resources/ghs-pictograms)

Perhaps an interesting detail to add: Humans are incredibly strong in recognizing human faces. This ability goes so far that we recognize faces in all kinds of natural shapes, such as clouds or trees, as shown in Figure 7.35.

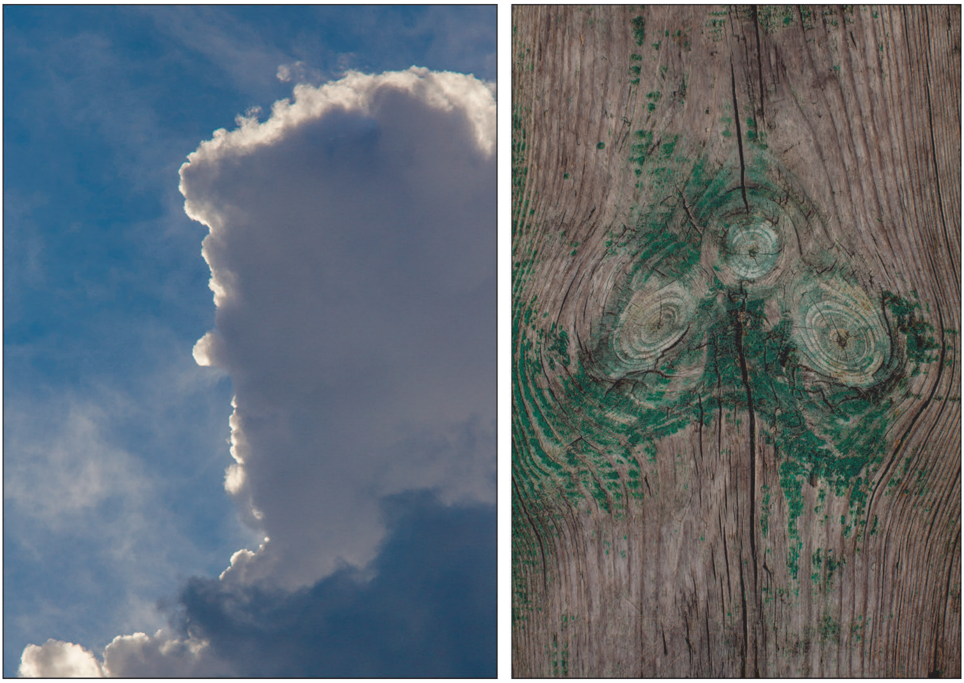


Figure 7.35 We recognize human faces even when no human face exists.

This focus on human faces makes a lot of sense. It helps us to recognize faces quickly and, above all, to get an impression of faces within a short time. We need less than a second to form a *first impression* from which, as we all know, there is no second chance. This quick processing helps us distinguish between “good” and “bad”—an essential skill for survival. Although fortunately we rarely struggle to survive in our part of the world, the ability to do so is still present.

7.3.5 Decide

Once you’ve discovered and recognized the stimuli, information processing is far from over. Now comes a crucial point that is sometimes completely overlooked in the design of UIs. We are talking about the *decision*. Just because you’ve detected something doesn’t mean you’re compelled to respond in a certain way.

Think back to the earlier example in Section 7.3.4 of the green traffic light: You have discovered the traffic light and that the green light is on now. You have had your driver’s license for quite a while and know that green means you can drive. Nevertheless, at that point, you might decide to stop. You may also notice that another vehicle is approaching the intersection from the side without braking, or you may see that three cars are already in the intersection because a backup has formed. So, you always have the option to choose how you want to respond to a particular stimulus. One of our favorite quotes on the subject is attributed to Viktor Frankl, a well-known neurologist and psychiatrist:

> *"Between stimulus and response lies a gap. In that gap lies our power to choose our response. In our reaction lies our development and our freedom."*

What Frankl is trying to say is extremely important: It is not the stimulus that determines what we do, but our decision and our reaction. Let's illustrate this relationship with a few examples. We start harmlessly and then escalate into the dramatic.

Imagine you're driving in your vehicle toward the intersection again. This time, however, the traffic light is not green but yellow. What do you do? Maybe you stop, perhaps you continue driving. There are at least two (actually, of course, many more) possibilities. Since not all people, and probably not even you, always show the same behavior at a yellow traffic light, you already know that it is not up to the traffic light how you decide.

Another example: Imagine you're out on a walk with your children when suddenly it starts to rain. Within seconds, you're soaked to the skin. What do you do? Maybe you get angry, try to get your kids home faster, and rant about the bad weather all the way home. Or perhaps you dance with your kids in the rain, jumping through the puddles and loudly singing, *"I'm singing in the rain."* After all, you're already wet! No matter what you decide—there are different reactions. This example also shows: It's not the rain but your decision that matters, as shown in Figure 7.36.

An incredible amount could be said on the topic of decision-making. For you as a usability and UX designer, the following is essential to understand:

- A person may have recognized something correctly. But that does not automatically mean that they are making what you consider to be the right decision.
- A person's decision is strongly influenced by their experience (memory comes into play in this context again). It may be that the person has learned what is good and what is less suitable and decides based on this experience. But it can also be that the person has not learned what is good or bad and decides anyway.
- People always decide in the way that makes the most sense for them at that one moment. No human wants to act stupidly, but they decide what they are capable of at that moment.

What does this mean for you now? If the users of a system, product, or service must make the "right" decision, then you must help them. Ideally, you would only let users make the necessary decisions anyway. If you can automate something, do it. The assumption is that every consciously made decision costs energy and that the amount of energy a person can devote to decisions per day is limited. Additionally, mistakes can occur with any decision. So, try to limit the number of decisions to what needs to be made.

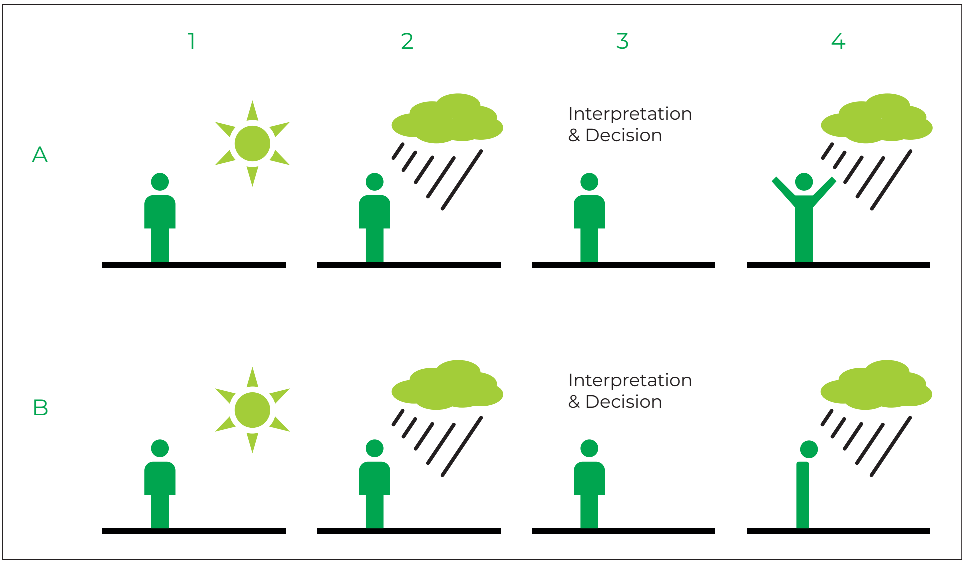


Figure 7.36 Two different decision options when it rains: A: Cheerfully sing "I'm singing in the rain" or B: Get angry and go home wet.

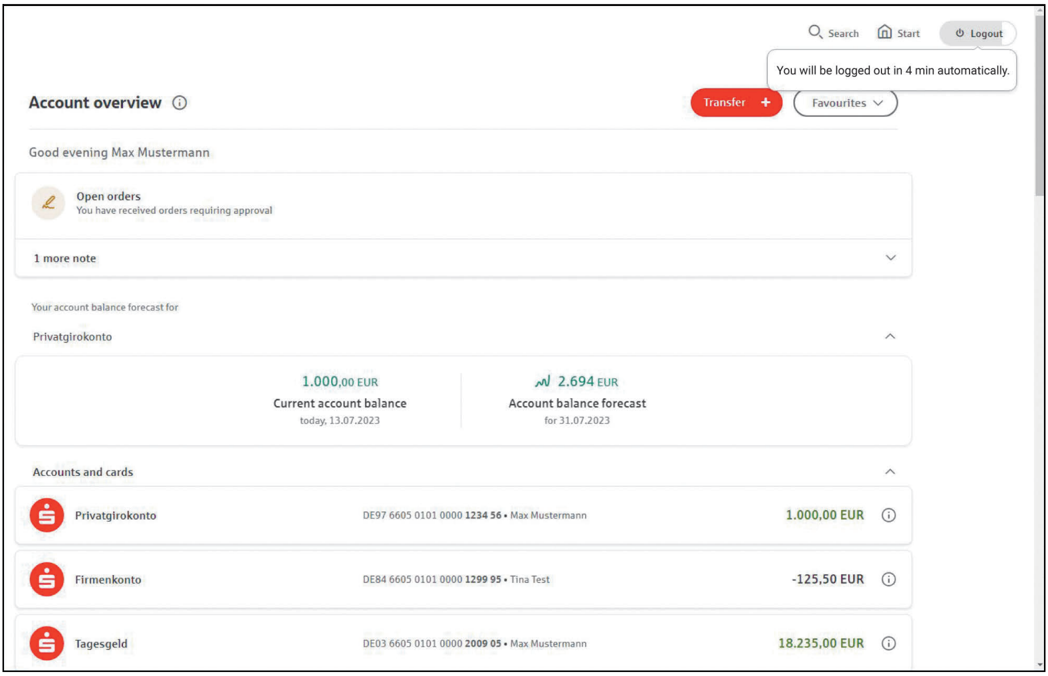
Also, give your users the information they need to make these decisions. If you don't, a decision will be based on the available information. If this information is not sufficient or incorrect, this failing will be reflected in the decision. Also, remember that information retrieval is poorer, especially under stress, and users tend to fall into habitual patterns. So, support decisions, especially in stressful situations.

Let's look at some examples that further emphasize the link to usability and UX design.

Imagine you're designing an online banking application. In this application, users are automatically logged out after some minutes. You also present this countdown with a timer in the upper-right corner, as shown in Figure 7.37.

So now, when your users see that they are still logged on to the online banking interface for 4 minutes, they may decide to pick out the appropriate bill to transfer quickly. Once they find the bill, they realize the logon has expired and must log on again. Or they might take another action just before the deadline to start the timer again and stay logged on that way. What users decide to do thus depends on whether they believe they can still complete an action in time.

Something similar could happen if a device's battery gauge shows only 10% charge left. The user discovers the information and recognizes that 90% of the battery is empty. However, whether the user subsequently looks for a charging cable depends quite a bit on how long the 10% battery usually lasts. If severe consequences could arise if the user turns off the device due to a low battery, then you should support the user in making a good decision. For example, you could provide an estimate of the battery's time to last instead of giving a percentage charge. Or you could add a warning message when the battery level drops below 5%.



You will be logged out in 4 min automatically.

Account overview

Good evening Max Mustermann

Open orders
You have received orders requiring approval

1 more note

Your account balance forecast for
Privatgirokonto

1.000,00 EUR
Current account balance
today, 13.07.2023

2.694 EUR
Account balance forecast
for 31.07.2023

Accounts and cards

Privatgirokonto	DE97 6605 0101 0000 1234 56 • Max Mustermann	1.000,00 EUR
Firmenkonto	DE84 6605 0101 0000 1299 95 • Tina Test	-125,50 EUR
Tagesgeld	DE03 6605 0101 0000 2009 05 • Max Mustermann	18.235,00 EUR

Figure 7.37 Online banking (Sparkasse) with a timer in the upper-right corner.

Numerous ways exist to help you help users make what you think are good decisions. So, if it's important to you how users decide to interact with your system, product, or service, support them as much as possible. To summarize once again: In the decide step, the user takes all the information they can reasonably get and chooses from the various options. So, when the step is "completed," the person knows what they want to do.

7.3.6 Act

The final step in the model is the actual action. Now, you may ask yourself, "Yeah, how? It's already decided what's going to happen here!" Yes, you're right. But a difference exists between deciding and actually doing something.

Imagine again that you're driving your car toward the intersection. The traffic light jumps to yellow. You *discover* the color change, *recognize* that it means the light is about to turn red, and *decide* you would rather drive through the intersection than brake. You then step on the gas to accelerate faster. Unfortunately, you're not driving your own car today but a rental car. Instead of hitting the accelerator pedal, you fully press the brake pedal and stop well before the intersection after full braking. So here, the decision and the actual action contradict each other—in this case, because an error occurred in the implementation.

Sometimes, however, the decision and the action performed afterward differ because external influences come into play. Imagine, for example, that you manufacture a cleaning agent. Your cleaning agent has a cap with a safety device to prevent children from accidentally coming into contact with the toxic substance. However, it just so happens that your development department is staffed exclusively by men who lay out the prototype (i.e., determine how it will ultimately look and function). When you first throw the product into a test with users, you notice that some women can't open the cap because they don't have enough strength. This problem exists at the act level—even though the decision to open the bottle has been made.

Thus, in the *action* stage, the important thing is to implement the decision made in the real world. As we have seen from the two examples in this section, the situations in which users cannot perform the desired action are the most exciting in usability design. The reason: there is still room for improvement for a better design. Therefore, let's discuss a few more examples where the decision and the action differ.

Imagine you want to step out of a building into the fresh air. You look around for the exit door, discover it, and recognize it as an exit door. You decide to open the door by pushing, which you try. As you perform the act of pushing, you discover a counterpressure. You recognize that the door can be opened by pulling, and you decide to pull on the door. Finally, the act of pulling leads to the desired opening of the door. In other words, ignorance of the situation (the door can be opened by pulling) initially led to incorrect action.

The environment itself can also influence the action. Imagine the intersection situation once again. This time you decide to brake, and the action execution is also correct because you press the brake pedal. However, since it has rained heavily beforehand, your car begins aquaplaning—and thus, you experience a different result, although you have done everything correctly on the action level.

The examples we've given so far have been played out in physical products, but errors at the action level also occur in digital products. For instance, you want to reply to a text message on your smartphone. You have already read the message and thought about a response. The first letter you want to press is an "I." You discover the "I" on the keyboard and recognize that it is the correct letter. You decide to press that key. However, as you press the key, you accidentally hit a key next to it.

A final example: You want to save something in an app on your smartphone, but don't read it and press the top-right corner because the save button on your smartphone is always in the top-right corner. This time, however, it doesn't say **Save**, but **Cancel**.

As we've warned before, our examples are simplified representations. With the aquaplaning example, we could now discuss whether the decision was made correctly. With the cap example, we could say, "The action was right, but the result was wrong" (i.e., the cap was turned in the right direction, but the bottle was still closed). And with the saving one, you could say, "There was already a perception problem." You're right about that. The model is designed to help you discover gaps in information processing. So, you can always discuss where you see a mismatch and adjust your solution. Towards the end of the book (Chapter 12, Section 12.22), we'll provide some tips on this topic. However, another part of the model can help you a lot as a designer: the reaction.

7.3.7 Reaction

The *reaction* is your feedback loop. Remaining in the model, you have acted, and through the reaction, you receive feedback on whether the whole thing was successful. In the case of aquaplaning, you applied the brakes, which caused the vehicle to slide. At this point, the following information processing process starts. The stimuli relevant to this example would be mainly visual and vestibular. The decision and action occur again if the stimuli are detected and recognized correctly. After that, the feedback loop starts all over again.

Note that memory is also used during each run—on the one hand, to provide the necessary information, but also to learn from the situation and how it arose.

Since memory is essential to us as designers, anyway, let's go straight to it and look together at what memory can do.

7.3.8 Memory

Memory is really complicated. As always, we'll try to keep it as simple and as applicable as possible in this book. We've summarized all the relevant information about memory in this overview section.

First, *memory* doesn't exist as a single concept. In fact, you have different types of memory that work together to help you remember things, learn things, and remember special moments years later. Put simply, you have three different types of memory, as shown in Figure 7.38.

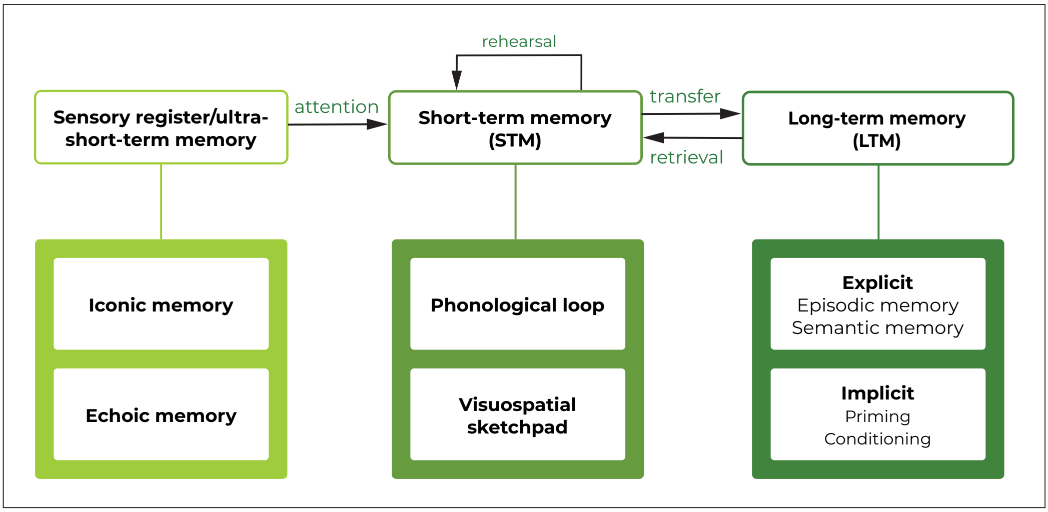

rehearsal
Sensory register/ultra-short-term memory
attention
Short-term memory (STM)
transfer
retrieval
Long-term memory (LTM)
Iconic memory
Echoic memory
Phonological loop
Visuospatial sketchpad
Explicit
Episodic memory
Semantic memory
Implicit
Priming
Conditioning

Figure 7.38 Three types of memory.

The Sensory Register

The first kind of memory is the what's called the *sensory register*, which is more or less considered the last part of information processing. Our focus in this context is the memory that allows you to filter out the relevant ones from infinite stimuli. Sometimes, the sensory register is also called *ultra-short-term memory*.

In the sensory register, all stimuli that come to us are stored temporarily. Because the abundance of information is so great, this intermediate storage is ultra-short. When we say "ultra-short," we mean a duration of 0.5 to about 2 seconds. Everything is stored first in this process, regardless of whether we are currently focusing on it. Maybe you even know ultra-short-term memory from your own experience: Have you ever had a telephone conversation in which you didn't understand part of a sentence and asked directly? Perhaps, when the other party starts talking, you immediately remember what was said. You hear that part of the sentence again, so to speak. This scenario illustrates how your *echoic memory* means you can recall the sentence again.

The Short-Term Memory

Via conscious attention and concentration, elements from the sensory register then move into your short-term memory, as shown in Figure 7.39. Everything that does not receive attention fizzles out again and is forgotten. What's called a *rehearsal* then happens in the short-term memory in which the information is consciously repeated or, for example, mapped visually.

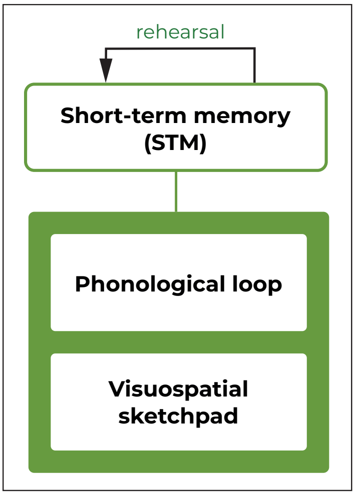


Figure 7.39 Short-term memory.

Perhaps you know that moment when someone tells you their phone number, and you then search for a pen and recite the number over and over again—more or less silently—so as not to forget it. The number is probably gone when the phone rings and you answer it. The process of reciting over and over again is the rehearsal, which itself is very prone to glitches. When a glitch occurs, the information is usually replaced directly by new information.

Unfortunately, the amount of information stored in short-term memory is relatively limited. In general, the short-term memory can hold approximately ± 7 units—so-called *chunks*. However, this is also a great opportunity! Memory masters manage to group information differently and thus remember significantly more content. The trick is to form one big chunk out of several small units. Try to remember the following numbers:

07041776

If the numbers are simply numbers to you, then remembering 8 digits may be the limit of your short-term memory. However, if you realize that the numbers are the day the US proclaimed independence from England (July 4th, 1776), you only need to remember one piece of information. In this way, you can stretch the limits of short-term memory and remember more information in a short time.

Methods also exist to train your memory by remembering things spatially. Let's say you imagine your apartment and then arrange the things you want to remember in your apartment. For example, the elephant hangs on the coat hook at the entrance. The giraffe is on the shoe rack. With these methods, you must see things in front of you and have a set path through your home (or another place) that you keep going. So, your short-term memory is limited, but it can be tricked to grasp more information.

The primary function of our short-term memory is to serve as an intermediate store. Most information remains in short-term memory for no more than 30 seconds before it is transferred to long-term memory, as shown in Figure 7.40.

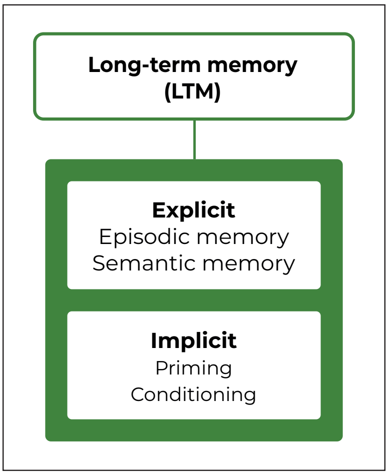


Figure 7.40 Long-term memory.

The Long-Term Memory

As with all other transitions in memory, there is also a kind of bouncer called the *limbic system* that looks at the information and evaluates it. All information that evokes emotional reactions (joy, fear, anger) has a greater chance of getting past the bouncer than neutral information. The stronger the emotions, the more likely we are to remember something. For instance, sometimes we have a hard time learning things that don't interest us but that we need to learn for an exam.

Against the backdrop of thousands of years of harsh history in an inhospitable environment, our reliance on the limbic system makes perfect sense: Our brain remembers what we need to survive positively and negatively. On the one hand, positive emotions make us repeat things (so we eat the sweet berries again). On the other hand, negative emotions make us stop doing things, not repeat them in the future, or not start them in the first place. In this way, we avoid dangers that could become dangerous for us and our survival.

About Forgetting Information and the Jungle Metaphor

The capabilities of our brains are impressive. Researchers believe that—except in cases of illness or injury—no information in the brain is ever forgotten once it makes it into long-term memory. Does that strike you as odd because you're sure you always forget things?

Imagine a dense jungle with many tall trees and thick bushes but virtually no open space. This jungle is your brain. Whenever you store a new piece of information, a new tree or shrub is created somewhere, which you've planted. In the beginning, remembering the plant is easy because you're standing right next to the tree: After all, you just planted it. At some point, however, it turns to a new tree, and then you must find your way back.

You can imagine the retrieval of the information as if you were a ranger beating a path through the impassable undergrowth with a machete. A lot of effort is required, and the way is not easy. This same process occurs when you must read or learn individual pieces of information over and over again. You have beaten a path to the tree when you first remember it. If you recall the information repeatedly in the aftermath, the path becomes wider and easier to walk each time. You'll progress faster until you don't have to think about it anymore, and you'll be roaring down the highway to your tree.

Then, at some point, when the information is no longer important, you retrieve the information less frequently. The highway slowly begins to narrow again and eventually grows completely closed. The tree is still there, but it's much harder to find your way. Nevertheless, you may still see a small clearing where the path once began. You'll notice that refreshing knowledge that already existed is usually much faster than learning it completely new.

The jungle metaphor also illustrates quite well why it is easier to remember knowledge that can be linked to existing knowledge. Imagine that a path and a clearing in the trees already exists in this jungle and that you only need to enlarge the clearing a little or only need to add a little additional path because some resources are already available.

In general, long-term memory is subdivided into explicit (or declarative) memory and implicit (or non-declarative) memory.

Explicit memory includes everything you're aware of and can name explicitly—in other words, everything that is linguistically accessible to you. On the one hand, this consists of *semantic memory*, where factual knowledge is stored. (Remember: July 4th, 1776, is Independence Day in the US.) In addition to the semantic memory, we also have *episodic memory*. All the facts and memories about your personal life are stored here.

Implicit memory stores everything that could be described as unconscious knowledge (in technical jargon: *implicit knowledge*), which includes *procedural knowledge* (knowledge about specific processes and actions like riding a bicycle), priming, and conditioning.

Conditioning includes all learned responses to a particular stimulus, for example, that your mouth waters at the sight of your favorite food or you automatically grin when a camera is pointed at you.

Priming is when the processing of a stimulus is facilitated by the fact that another associated stimulus is sent in advance. You may have personal experience with this: Sometimes, several advertisements from the same manufacturer come in short succession in one advertising block. The first one is the long version, and the others are much shorter. The shorter versions intend to draw your attention to the message of the advertisement and to make processing easier.

For your work as a usability and UX designer, the contents of *explicit memory* will be decisive in most cases. Explicit memory includes, for example, learned warning symbols or technical terms. However, *procedural knowledge* will also be vital for you in some cases.

For example, we once had a funny experience when Michaela tried to log on to Dropbox for about 20 minutes, getting angrier and angrier. At the time, Michaela usually logged on to Dropbox daily and knew the logon process inside and out. However, nothing was working this day, and she desperately needed a file stored on Dropbox. Again and again, she called up the website and clicked through the interface until, at some point, she sat in front of the computer cursing loudly (very unusual). When Ben came to look, it turned out that Dropbox had changed the logon page, and now—although the page was completely identical in structure—the logon field was no longer displayed, only a registration field. But Michaela hadn't discovered that because she knew how to log on.

So procedural memory can also play a role in design. And otherwise, you'll already have noticed that ten people have twelve different opinions, especially in design. In the next section, we want to look at where this comes from.

7.3.9 Conclusion

The human sensory system, memory, and information processing are powerful. So far in this chapter, we've provided an initial overview to take away the most important aspects. Above all, you should take away from this part of the book that the abilities of perception are relatively similar in all people, so you can consider these abilities and limitations when designing.

Note also that information processing follows a specific pattern, and that knowledge of this pattern will allow you to analyze the causes of errors and improve products.

Finally, be aware that memory dramatically influences our perception and decisions by sorting and evaluating all our experiences. In this way, the same stimulus leads to different associations and reactions in every person.

So, you can't design the people's experiences; you can only ever design the product. What the user makes of it, you can try to measure and optimize for many. But perception is subjective. Psychologists call this *constructivism*. Everyone creates their own worlds, which they believe is *the* real one.

In Section 7.3.4, we talked about recognizing human faces and described how the process takes less than 1 second. We assume that fast recognition also applies to products. Our brain has learned to judge based on appearance within the shortest possible time to survive in a dangerous world. Why should this be any different for a product? We assume that we also make an initial distinction between “interesting” and “not interesting” or “good” and “bad” within a short time in the case of systems, products, and services.

Several scientific experiments support these statements. Researchers at the Massachusetts Institute of Technology (MIT) have found that products shown for only 4 seconds show clear reactions in the brain (Knutson et al., 2007). In the test setup, subjects could decide after 4 seconds whether or not they found the product attractive and would purchase it. In another experiment by Lindgaard et al. (2006), 50 ms was determined as the time a web page has to make a good first impression visually.

Sometimes, just a few seconds is enough for the purchase decision. In other cases, you go deeper and try to compare more objectively. But let’s be honest: Do you also know the moment when a product wins the comparison because you want it to win? We do. So, in the next section, let’s see what you can do to make your product’s first impression as positive as possible.

7.4 Design Basics

This is the perfect segue to the topic of design basics. When you think about the effect of design, you quickly get to the point that individual taste plays a significant role. So, whether a design is liked depends a lot on the viewer. Following this point of view, it is an almost impossible task to inspire all or even many people with the same design. But this goal is often precisely the task when designing products—especially for larger business-to-consumer (B2C) markets. For example, Healthyfy, our health platform, faces the problem that the design must appeal to older and younger patients. But if the design is highly subjective, how can the founders of Healthyfy solve the task?

7.4.1 Impact of Design

Fortunately, design is not purely subjective. To a certain extent, the effect of design is even similar across different groups of people. For example, the “Maluma and Takete” experiment conducted by psychologist Wolfgang Köhler (1929) involved displaying the shapes shown in Figure 7.41 to test subjects. He then asked the subjects to associate with the shapes either the word *Maluma* or the word *Takete*. How about you? For you, which shape fits better with Maluma and which with Takete?

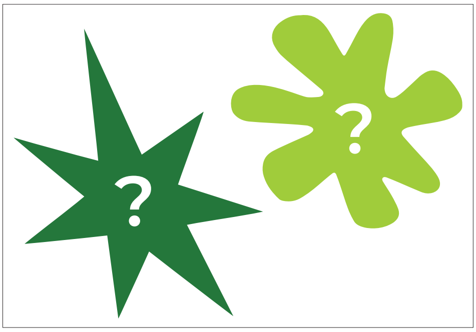


Figure 7.41 Maluma and Takete... or is Takete and Maluma?

If you answer 90% of the respondents, you have assigned the word Takete to the pointed shape on the left and the word Maluma to the rounder shape on the right in the picture. The experiment shows that the effect of design can be controlled in principle. Nevertheless, many influencing factors are at play, for example, the culture, the environment, and the viewer's previous experiences. That last point especially is highly individual and is where memory (Section 7.3.8) comes into play again.

But how do these two statements now fit together? On the one hand, we have established that the effect of design can be controlled in principle. On the other hand, this effect is highly individual. To answer this question, let's first take a closer look at what is understood by the effect of design. In this context, we are guided by Monika Heimann and Michael Schütz (2017), from which the core statements of the following sections are taken.

How Does Design Work?

When we look at a picture, no matter how complex, a dialogue occurs. The image triggers memories, ideas, everyday experiences, wishes, and dreams in us. These associations are individual because memories, imagination, and daily experiences also differ from person to person.

Design works through individually spanned association spaces.

The Link between Products and Emotions

The term *association space* is taken from Monika Heimann and Michael Schütz (2017). However, many other authors have also discussed the link between products and emotions. One of the best-known books is *The Design of Everyday Things* by Donald Norman, published in 1988. In this book, Norman assumes that there are three levels of perception: the levels of *visceral design*, *behavioral design*, and *reflective design*.

The level of *visceral design* describes the colloquial "gut feeling." People can consider a product beautiful and practical without interacting with it. This decision can also be made quickly and may lead directly to a purchase.

The level of *behavioral design* is primarily concerned with prior experience. For example, Norman shows that assumptions about usability and function can already be made based on experience without ever using a product. For example, products of a personal favorite brand get a leap of faith.

According to Norman, the level of *reflective design* is about conscious reflection. At this level, for example, the self-image of the human being comes into play.

These levels can be used to explain why a product can flop even though it has excellent usability—but is not attractive to users. It can also be explained why a product of a famous major brand that is not yet fully developed becomes a (commercial) success because it is perceived as quite attractive.

The strength of the effect depends on which memories, ideas, and everyday experiences are triggered. The design effect is also strong if these are associated with strong emotions. Conversely, associations related to weak emotions also produce a weak design effect.

Remember: The stronger the emotions associated with the triggered associations, the stronger the design effect.

For example, the image shown in Figure 7.42 might trigger memories of a beautiful vacation during which you encountered fascinating species of animals on an expedition. You enthusiastically recall the great soundscape of the jungle and the breathtaking photos you took.

Figure 7.42 A jungle.

But if you have never experienced this vacation and instead deal with the deforestation of the rainforest, the associations may be negative. Because the subject is close to you, you may even get angry at those responsible and unconsciously clench your hand into a fist.

How Are Association Spaces Used for Product Design?

In product design, a common practice is to design a product that ultimately pleases users, which usually means that the users should find the product beautiful. The *association space* allows for a different way of looking at things, which we think is good: The goal is not for a product to be beautiful. The goal is that the association spaces spanned by individuals in the target groups largely coincide with the association space you want.

Your product (and associated marketing) has an outstanding design when it can dock onto the associations you want and trigger strong emotions in that way.

If you manage that, then you have achieved a robust UX. We want to note that positive emotions are not always the goal. For example, an app collecting donations for a war-torn country would initially aim to make you concerned and appeal to your compassion. After all, the app operators want to collect as much money as possible to help.

How Can You Design the Association Space, and What Must You Consider?

As described earlier, you first need to identify which association space your solution should span in the first place. Specifically, you can brainstorm or use another creative method (see Chapter 4, Section 4.2.4). Analyzing products already on the market can also help identify the desired associations.

In any case, the goal is to establish concrete associations. For example, you might want your product to appear “sustainable and modern.” In addition, it should be recognizable as a product in “wellness and relaxation.” If you had agreed on these associations with all stakeholders, you would also already be done with the first definition of the association space.

Chapter 11, Section 11.12.2, will go into more detail about designing the association space. In this section, we'll look closely at what needs to be considered.

An essential aspect of the design of the association space is that the desired associations should match the product or its characteristics. Suppose there is a connection between the desired association and the product property. In that case, the product ideally links the memories, ideas, everyday experiences, wishes, and dreams evoked in the users with your product.

If the connection is missing, then the link is also missing, and the positive effect is missing.

Suppose you want to trigger associations in wellness, as described in this example. In that case, the easiest is to develop a wellness product with related properties. An app that revolves around cooking and recipes at its core could also work with the association of wellness. If the association is also clear to the users, then your design can tie into the experiences users have already had in wellness.

However, if the recipe app focused on cooking training and coaching to become a master chef, then users would probably lack the connection to the topic of wellness; this connection can be tested by the *association space test (ARTTest)* described in Chapter 12, Section 12.19. The desired associations also fail to materialize due to the lack of a connection to the app's features.

An example of the often non-existent connection between product characteristics and associations is the phrase "sex sells." Contrary to the widespread opinion that this concept always works, this motto is often shown to be false. Only products that want to place themselves in this theme world also benefit from the associated associations. But if, for example, our customer relationship management (CRM) tool *BusinessBooster* tries to incorporate "sex sells" elements into product design and marketing, this decision will hurt the brand and the product. A better idea would be for *BusinessBooster* to create associations such as "security" and "faster business deals."

Unfortunately, the factors influencing perception and the association space are so diverse that no simple "if-then" rules are available, and the actual effect of a visual design can only be determined with certainty through surveys. Additionally, depending on the design solution, you may pursue different goals. For example, what is goal-oriented in a productively used work application may be a "bad" design in an explorative PC game that stimulates curiosity.

Nevertheless, general principles will guide you in designing your products and help you derive the best possible visual design with the desired effect. We'll get to surveys and testing your design later in Chapter 12, where we'll look in detail at the various methods you can use to see if you're on the right track. But in this chapter, we'll get you there first.

To this end, in the next few sections, we'll briefly introduce you to the essential elements central to UI visual design.

7.4.2 Basic Elements of Visual Design

A visual design is composed of various basic design elements, for example, colors and shapes. The selection and combination of the basic elements are central to the effect of the design on the viewer. When you design your solution visually, in simple terms, the task is to select from the basic elements and combine them with others that trigger the desired associations in the users.

We are aware that this description is still abstract and that the question "How exactly does it work?" remains unanswered at this point. We want to introduce you to the basic elements in this section before we take them up again later in Chapter 11, Section 11.12, when we create a concrete UI design.

When presenting the basic elements, we'll limit ourselves to the essentials since each topic can fill books, and we want to provide an overview first and foremost. We'll present the following basic design elements in the following sections:

- Colors
- Shapes and surfaces
- Lines
- Materials and textures
- Typography
- Images and graphics
- Icons

Each basic element alone can trigger the desired associations. For example, pastel shades of blue and pink are suitable for a wellness product. Typical images in the wellness area are nature images, for example, smooth stones, blossoms, and candles. The whole effect is created by combining the basic elements, for instance, if the flowers are in a pastel shade of pink.

Now, let's start with the first basic element: colors.

Colors

Humans take in most of the environment through their eyes. Colors play a significant role. Markus Wäger (2017) writes right at the beginning that there is no design without color. (He includes black and white as colors.) We see it the same way. The choice of colors is enormously crucial for a successful UI. But what is color?

If you have read Section 7.2.1 about the human eye, you know that colors are sensory impressions stimulated by light of specific wavelengths.

Colors—like any other basic design element—can trigger associations. In addition, colors can be used to direct users and draw their attention to dangers, for example. As usability and UX designers, we use colors (in combination with other basic design elements) to achieve just that. From our point of view, the following topics are central to usability and UX design:

- The selection of appropriate colors to achieve the desired associations
- The selection of suitable colors to support content-related messages contained in texts, icons, or buttons
- The contrast ratio between colors
- The consideration of color vision defects

Rather than including all color basics in this book, we picked out only some essential aspects for usability and UX design. For a basic introduction to colors, we recommend the books *Interaction of Color* by Josef Albers or *The Secret Lives of Color* by Kassia Clair.

Selecting Colors that Achieve the Desired Associations

Heimann and Schütz (2017) describe how the effect of color depends strongly on connections with other elements. For example, red can stand for love or eroticism but also for aggressiveness and danger. Depending on the context, therefore, a different effect results.

Conversely, the color can give a context a different effect. For example, a red rose as a gift has a different meaning than a white rose. Culture also plays a significant role. In China, for example, the color red is a symbol of joy, happiness, and prosperity and is therefore represented at many festivals in red ornaments, lanterns, or clothing. (Especially at weddings: A traditionally dressed bride wears a red dress.)

Colors can also be used to create affiliations to groups. For example, we classify babies in light blue and pink. If you dress a male infant in pink, Western culture assumes the child is a girl. Subcultures such as punk or goth also have specific colors as a symbol of recognition and certain other trademarks. Colors can therefore create a sense of belonging.

The same thing that applies to subcultures also applies to some industries. In specific sectors, defined industry codes have emerged as color worlds. For example, organic food brands and designs are often in shades of green and brown. The medical industry often uses a combination of white and blue or white and turquoise. You can probably think of other examples.

When you choose colors for the design of your solution, you should be aware that colors can create different effects depending on the context. Be mindful that the color scheme could signify belonging to a group. This association allows you to control how your design is received. Do you want your solution to connect to an already established group? Then, you should use the group's colors.

Viewers then have the relevant associations and can directly identify your product (e.g., as a medical device). On the other hand, you could stand out with a completely different color choice. The target group may not immediately recognize a pink device as medical. But you'll stand out.

In summary, you should consciously choose the colors for your solution to trigger the desired associations with other basic design elements. We'll show you a procedure for making this choice in Chapter 11, Section 11.12.

Selecting Colors to Support the Content Message of Texts, Icons, or Buttons

Once you've determined the primary colors of your application, you need to move on to the next challenge: deciding which colors to use for what.

Usually, in digital applications, you have a *call-to-action (CTA)*, which is a request for the user to do something specific. This action can be, for example, submitting a form. Usually, CTAs have a color that goes well with the color language of the application but is perhaps a bold version of that color. For example, in a blue-toned application, the primary CTA might be an intense blue, as shown in Figure 7.43.

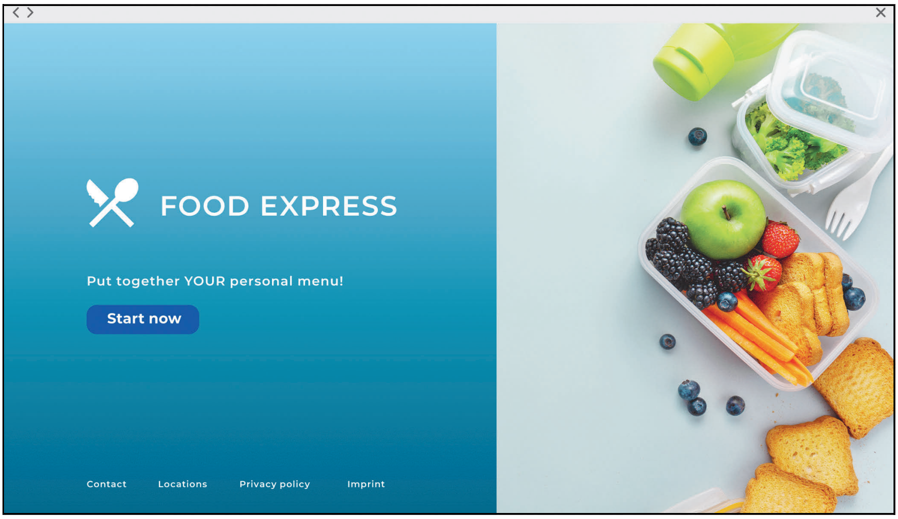

FOOD EXPRESS
Put together YOUR personal menu!
Start now
Contact
Locations
Privacy policy
Imprint

Figure 7.43 Sample application in shades of blue, with the primary CTA in an intense blue color.

However, you can also highlight the CTA with a contrast color that does not appear anywhere else on an interface, for example, a yellow CTA button in an application that is otherwise mostly blue, as shown in Figure 7.44.

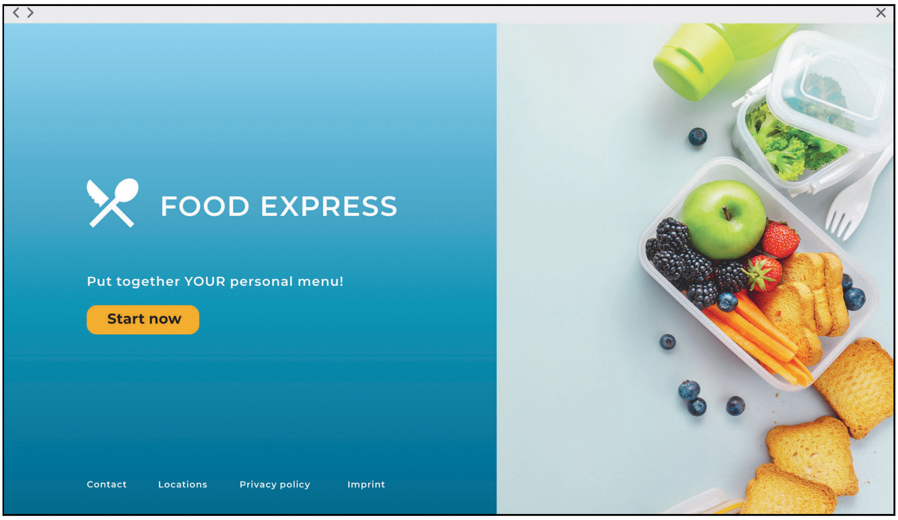

FOOD EXPRESS
Put together YOUR personal menu!
Start now
Contact
Locations
Privacy policy
Imprint

Figure 7.44 Sample application in shades of blue, with the primary CTA in a contrasting hue.

The design of the primary CTA usually still works quite well, but secondary buttons are often designed in a very subtle color. In our example, gray tones are regularly used so that the button fades into the background compared to the primary CTA. Depending on the design, however, a false impression can quickly arise here because, unfortunately, gray in the interactive world often means “grayed out” and thus “inactive.” So, what matters here is finding an in-between path between visual and functional design that enables an appealing impression and good usability.

If you find it challenging to identify matching colors, there are now several tools to help you, such as *https://colorizer.org*, as shown in Figure 7.45. The *Colorizer* helps you to find matching colors based on a selected base color. In addition, you can also look for matching color triads or complementary colors (the color at the opposite end of the color spectrum).

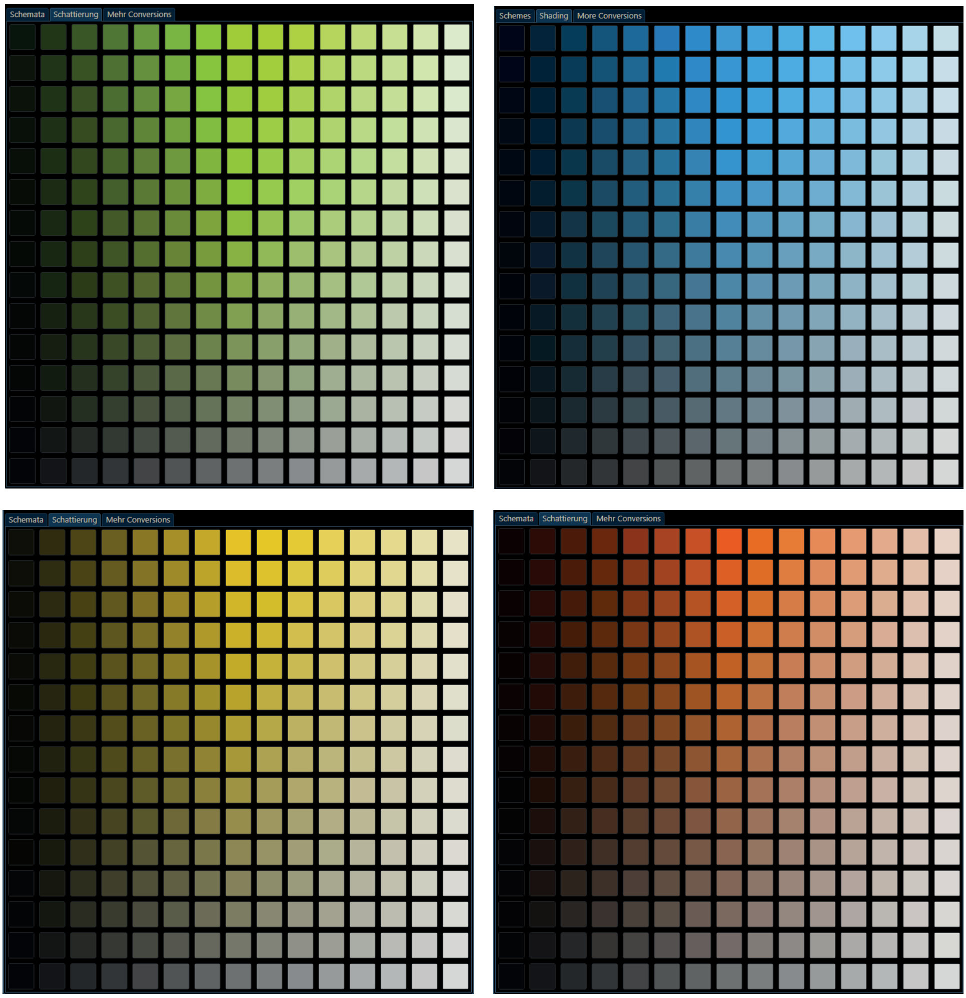
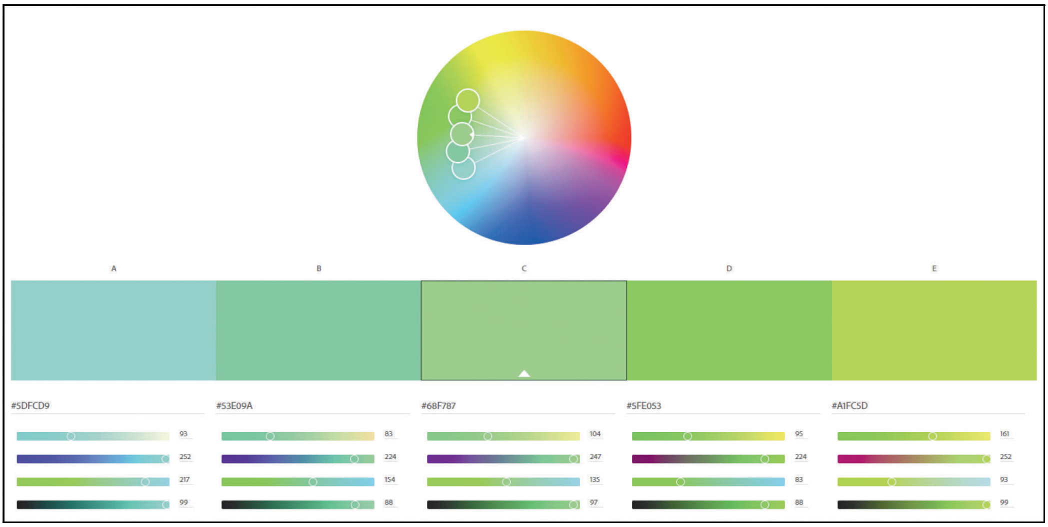

Schemata
Schattierung
Mehr Conversions
Schemes
Shading
More Conversions
Schemata
Schattierung
Mehr Conversions
Schemata
Schattierung
Mehr Conversions

Figure 7.45 Compiling different color palettes (https://colorizer.org).

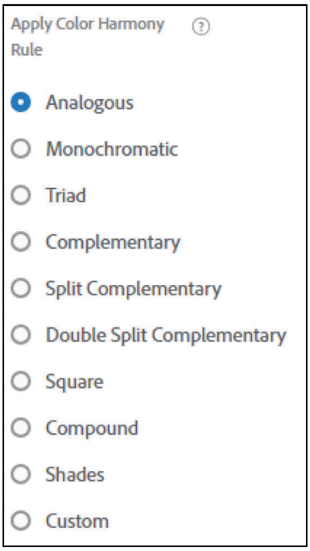
Besides the Colorizer website, other websites can help you choose colors, such as *https://color.adobe.com*, where you can select suitable colors based on a color wheel, as shown in Figure 7.46. In addition to color selection, both tools support you by providing the respective color values in different color modes, such as RGB or CYMK, and as hex codes.



A	B	C	D	E
#5DFCD9	#53E09A	#68F787	#5FE053	#A1FC5D
93	83	104	95	161
252	224	247	224	252
217	154	135	83	93
99	88	97	88	99

Figure 7.46 Adobe's color selection tool helps you choose harmonious color values for your solution (https://color.adobe.com/create/color-wheel).

In addition, various color harmony rules can also be applied in Adobe to combine the colors, which subsequently provide different colors as a result, as shown in Figure 7.47.

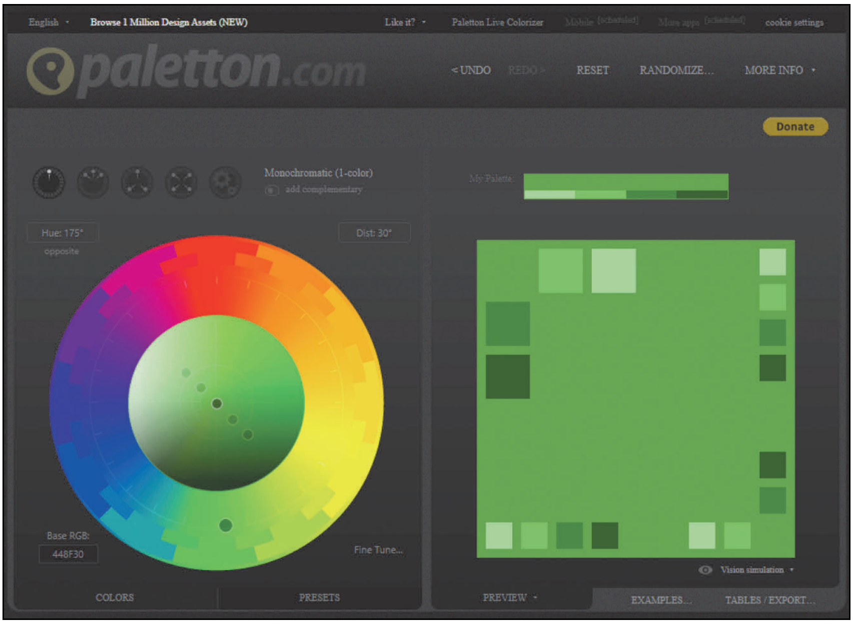


Apply Color Harmony Rule

- ● Analogous
- ○ Monochromatic
- ○ Triad
- ○ Complementary
- ○ Split Complementary
- ○ Double Split Complementary
- ○ Square
- ○ Compound
- ○ Shades
- ○ Custom

Figure 7.47 Selecting color harmony options (https://color.adobe.com/create/color-wheel).

Of course, other providers exist, such as *www.paletton.com*, as shown in Figure 7.48. However, the functions of the various providers are quite similar across the different tools.


English
Browse 1 Million Design Assets (NEW)
Like it?
Paletton Live Colorizer
Mobile [scheduled]
More apps [scheduled]
cookie settings
paletton.com
< UNDO
REDO >
RESET
RANDOMIZE...
MORE INFO
Donate
Monochromatic (1-color)
add complementary
My Palette
opposite
Fine Tune...
Vision simulation
COLORS
PRESETS
PREVIEW
EXAMPLES...
TABLES / EXPORT...

Hue: 175°

Dist: 30°

Base RGB: 448F30

Figure 7.48 Color selection tool at www.paletton.com.

In addition to choosing a suitable color harmony, you should also consider the *content-related meaning of the information* that you want to color-code. In general, the brighter or more radiant a color is, the more attention this color or the object in this color gets.

It is, therefore, a good idea to consider in advance which elements on your screen should receive the most attention. These elements should then be given the colors that attract the most attention. These colors should be used sparingly, and ideally, there should be only one or very few elements in this color on each screen.

On the other hand, less important information or purely decorative elements should take a back seat so as not to distract from the critical content. From the point of view of usability and UX design, it is thus possible to weigh the content through the color design and guide users through the applications using the colors.

Tip: When to Break This Rule

The procedure just described is suitable for most UIs that support users in completing a task. However, there are also UIs that specifically address curiosity and aim at exploration by the users. For example, instead of highlighting a few elements with the color, a game could deliberately make many elements visually intense. In this way, unambiguity would be gradually lost, and users would be encouraged to explore.

In addition to prioritizing information, in some places, there is a need to make a statement about the content of a piece based on the color. This can be, for example, about the *criticality* of information or the correctness.

Especially in Western culture, there is color coding of green for all positive (correct, successful, saved, safe) and red for negative or critical information (wrong, error, not successful, dangerous). In some areas, there is also yellow for warnings and, if necessary, future critical aspects. In the web context, blue is often used for links, redirects, or information. You should at least be aware of this color coding to be able to use it selectively.

However, we do not recommend coding by color alone because of the possibility of color vision defects. As described in Section 7.2.1, you should not use color alone to explain a condition, for example. You could design positive feedback in green and additionally in the form of a checkmark.

We also advise against violating color coding (unless deliberately in a game, for example). For instance, if you display a positive confirmation, such as the successful sending of an e-mail in red color, this contradiction creates what's called *interference in the perception* (the overlapping of different meanings) of users. As a result, processing the information takes longer and is more error-prone than with a matching design.

Contrast Ratio

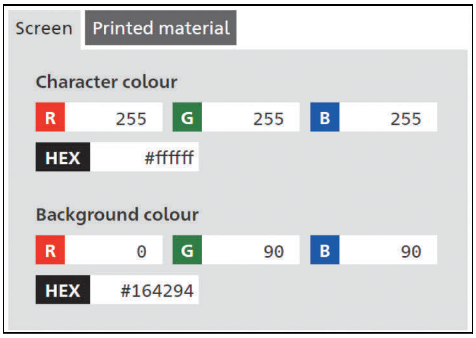
Contrast is about how a color stands out from the background color. Contrast can be created with the color of a font, but it can also be the color of an icon or button.

The contrast issue is often a stumbling block when transitioning from wireframes to visually designed screens. Whereas previously the work was mainly done with black and white screens, visual design suddenly brings color into play. This color is necessary to create the correct association. But the color is also essential to make the system, product, or service suddenly feel "finished." Nevertheless, all too easily at this point only color harmonies are thought of and designed, while aspects of usability and UX are forgotten. We want to give you the tools to prevent this problem from happening to you.

To determine whether two colors differ enough, several formulas can calculate the contrast between two colors. To calculate contrast ratios, you have the following options:

- Deutsches Institut Für Normung (DIN) 32975:2009-12 is a German norm that provides a formula that determines contrast based on surface luminance. The DIN standard was initially aimed at designing information in public spaces, such as signage or passenger information. Unfortunately, the norm is not available in English, but the online calculator we describe next uses this formula.
- For calculating digital contrasts, a more common practice is to use *Web Content Accessibility Guidelines (WCAG)* rules, which deal with accessibility in the digital space.

To make the calculation of the contrast ratio easier for you, we won't discuss the formulas in this book. Still, refer to the online calculator at *www.leserlich.info*, provided by the *German Federation of the Blind and Partially Sighted*. Despite the German URL, an English version enables you to conveniently calculate the contrast ratios of your designs online, as shown in Figure 7.49. You can find the contrast calculator at *https://www.leserlich.info/werkzeuge/kontrastrechner/index-en.php*.

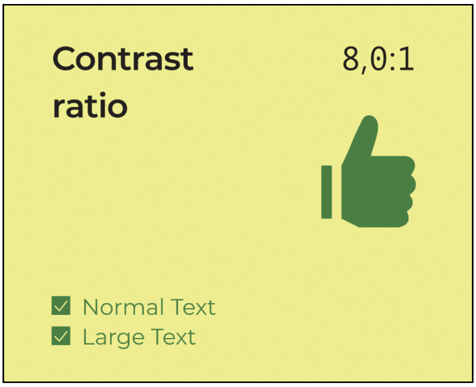


Character colour
R 255 G 255 B 255
HEX #ffffff

Background colour
R 0 G 90 B 90
HEX #164294

Figure 7.49 The contrast calculator from www.leserlich.info.

To perform the contrast calculation, specify the two colors involved, for example, the button color and the font color used on it. You can use either RBG values or HEX codes to represent these colors. You can also choose whether the display is a screen or for printed material. Afterward, you get an estimation of the contrast ratio on the right, as shown in Figure 7.50.



Contrast ratio 8,0:1

☑ Normal Text
☑ Large Text

Figure 7.50 Result of a contrast check on www.leserlich.info.

On the one hand, you receive a value for the contrast ratio, and on the other hand, an assessment of whether the value is excellent or poor. For the evaluation, a distinction is also made between large text (18 points and larger) and "normal" text (smaller than 18 points) since smaller text requires a higher contrast ratio to be easily readable, as shown in Figure 7.51.

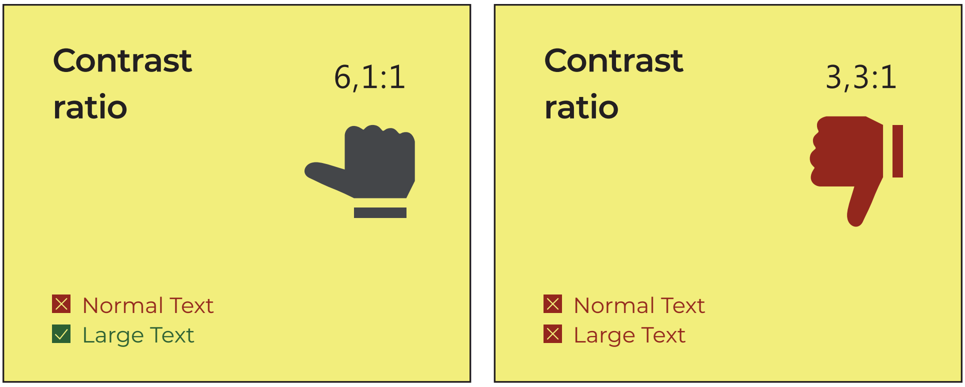


Figure 7.51 Two more results. On the left, everything is fine as long as you provide large text. For normal text, the contrast ratio is no longer sufficient. On the right, the contrast ratio is too low in any case.

Color Vision Deficiency

We touched on the topic of color vision deficiencies when we discussed the basics of perception earlier in Section 7.2.1. Let's briefly review some essential facts about color vision deficiencies:

- Different color vision deficiencies exist, the most common being red-green deficiency.
- Yellow-blue deficiency also exists.
- Some individuals suffer from complete color blindness.

Therefore, if you want to choose an excellent visual design and still make sure that the usability and UX does not suffer, then you should:

- Avoid a visual design based solely on the colors that can cause perceptual problems.
- Always use an additional visual component—for instance, a cross, a checkmark, text that says "OK," or a thumbs up as earlier—in addition to purely color coding of information—for instance, red for "wrong," "error," or "dangerous" and green for "correct," "done," or "continue."

Color Space

Color spaces can best be imagined as a multidimensional space in which all the colors that this color space encompasses are represented as individual points.

The best-known color space—and most relevant for the digital domain—is the *RGB color space*, in which colors are specified as a mixture of red, green, and blue. The RGB color space is an additive color mixture in which "no color" means black. The different color tones can be created by adding additional colors, as shown in Figure 7.52.

Photographers primarily use the RGB color space to create and digitally display images and digital graphics. A typical RGB color space creates up to 17,000,000 (17 million) different hues on digital devices (Wäger, 2017). In addition, special RGB variants exist, such as the *Adobe RGB color space*, the *ProPhoto RGB color space*, or the *sRBG color space*, which was specially developed to offer identical color reproduction on many different digital devices. In this color space, up to 1,000,000,000 (1 billion) color tones exist (Wäger, 2017).

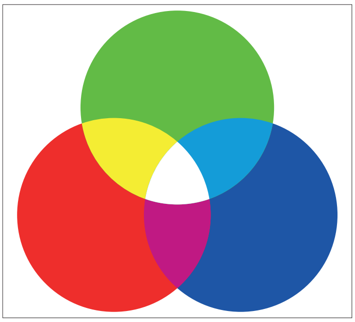


Figure 7.52 Additive color mixing.

In *printing*, color is no longer a matter of transmitted light (where the light is behind the object and shines partially through it) but instead a matter of reflected light (light is in front of the object and illuminates it). In this scenario, we are working with a different color space, namely, the *CYMK color space*. CYMK stands for cyan, yellow, magenta, and key (for black).

CYMK is a subtractive color mixing process. In subtractive color mixing, “no color” means white (as opposed to additive, where “no color” means black). Colors are filtered out of the white light as with a filter. If you filter out all colors, you get black—that is, no light, as shown in Figure 7.53.

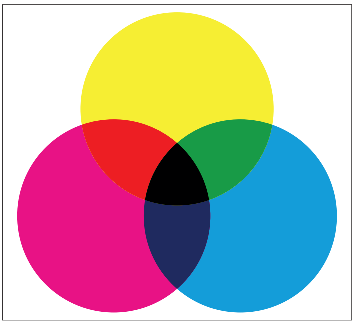


Figure 7.53 Subtractive color mixing.

In the meantime, extensions for the CYMK color space are no longer based on only four colors but use additional colors to create more colors in print. One example of such an extended color space is the *Extended Color Gamut (ECG)* color space, which contains orange, green, and violet.

Color Space for Digital User Interfaces: What Must You Pay Attention To?

The representation of colors based on the color space does not necessarily lead to the same result on all devices. For this reason, *calibrated* and *color-accurate monitors* are often used in the print industry and areas where high color fidelity is essential. The problem with needing high color fidelity is that, with digital UIs, you cannot assume that the monitors on which your design is displayed will also be color accurate (unless, by chance, you design an application for precisely this area). Depending on the solution you design, your product will run on conventional monitors or smartphones, for example. Therefore, when choosing colors, you should test the colors on the actual target devices as well, which is easy if you're designing for a few devices. However, if the group of devices you design for is going to be significant, you probably won't be able to test them all due to expense. For example, testing millions of different devices is impossible if you're designing a website that will later be displayed on all smartphones, tablets, and PC monitors.

Our recommendation is to select a representative set of devices and test your design on them regularly. Pay attention to color reproduction ("Is the green the way I want it? Does it communicate the information I want correctly?") as well as contrast ("Is everything legible? Do areas blur into each other that are supposed to be distinct?"). If you have equipment in this set that can display only a few colors (for example, industrial facilities are commonly equipped with monitors with a small color space), use only this reduced color space.

In addition, the RGB and CYMK color spaces differ from each other. So, if you have a template in print, you must assume that the colors will be slightly different on the screen. In the same way, creating a digital style guide for printing makes sense if the CYMK color space is already used during creation. So, plan during the creation on which media your design will be displayed and choose the appropriate color space.

Depth Effect

Besides the meaning of colors in terms of content, colors also have a *depth effect*. Usually, blues seem more distant to the viewer, and reds closer. Yellow is perceived as farther away than red but closer than green. Green, in turn, feels closer than blue. Again, from this insight comes a logic for using colors in your UI based on the association you want to achieve. (For more information, see Wäger, 2017.)

Shapes and Surfaces

Besides colors, *shapes* are one of the basic elements of design. Almost all shapes can comprise the basic shapes *dot*, *line*, *triangle*, *rectangle*, *circle*, and *cross*. Even a cross combines different lines or rectangles, a square is a particular rectangle form, and a dot is a tiny circle. And even a line is a rectangle with a mainly flat side. On the whole, you could say that you can make all shapes out of rectangles, circles, and triangles. Whether a shape looks more dynamic or balanced often depends on the relationship of its axes to each other. Perfectly symmetrical shapes look more stable and less dynamic than “slanted” shapes or those as wide as tall.

According to Heimann and Schütz (2017), on whose execution this section is based, shapes can be divided into different *orders*. The shapes you use can be represented directly; in which case, they are *1st order shapes*. But shapes can also be created by overlapping (*2nd order shapes*) or by arranging shapes around each other (*3rd order shapes*).

The 1st order shapes are usually used intentionally; the 2nd order shapes are created only by the interaction of different shapes with each other. For example, if you let two triangles overlap on one side, a third small triangle will be created in the intersection, as shown in Figure 7.54. However, 2nd order shapes can also arise when the arrangement of different shapes causes one shape to suddenly emerge from the background because the 1st order shapes in the foreground cover the other areas.

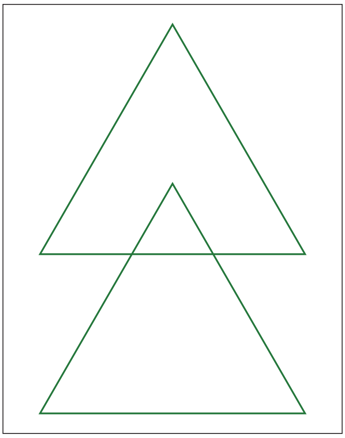


Figure 7.54 A triangle formed as a 2nd order shape in the intersection of two triangles.

In the case of 3rd order shapes, the perceived shape is created because all other shapes and elements have been arranged together to form a shape. An example might be the arrangement of different shapes on an imaginary circular line so that a circle becomes visible as an overall shape, as shown in Figure 7.55.

The *rectangle* is one of the basic shapes in design. Due to the straight sides and right angles, it looks static and consistent. The rectangle stands for order and structure. It can thus be found in other places, such as an element in a grid that helps you align and structure UIs. Rectangles are often used to mark boundaries. Rectangles are also often used in the design of UIs to delimit different areas from one another.

The *square* is a particular case of a rectangle and looks even more static and less dynamic because of its equal-length sides. So, if you want to express excellent stability and a bit of heaviness, the square could be a good shape for you.

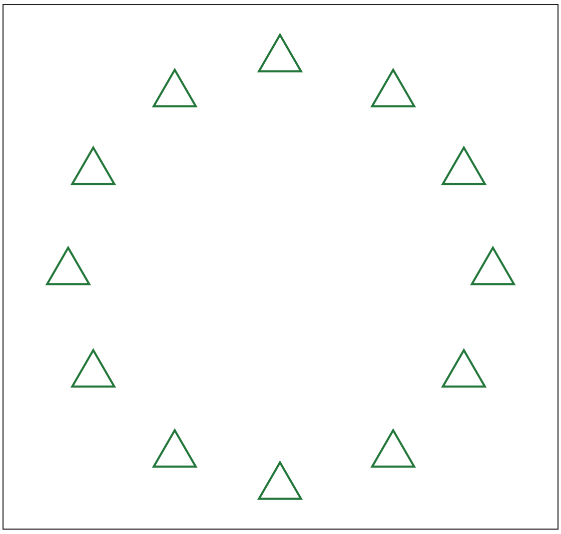


Figure 7.55 A circle emerging as a 3rd order shape by arranging other shapes on a circular line.

Compared to the rectangle, the *triangle* looks more dynamic, sometimes even aggressive (depending on how pointed you make it). The more pointed the triangle becomes, the more dynamic and aggressive it appears. The more balanced the side lengths of the triangle are, the more stable and less dynamic it appears. In a more pointed version, triangles can point like arrows in a direction, guiding the user's gaze. On the other hand, triangles are seldom used to delimit areas, as they offer little space, making larger areas unusable or usable only with difficulty.

The *circle*, on the other hand, is entirely balanced. It is not very dynamic and, as a standalone element, has no movement in one direction but rests in itself. Circles are often used to summarize content.

Even if the *dot* is only a particular form of the circle—namely, in theory, the smallest conceivable circle (in practice, the dot is small, but not the smallest conceivable to recognize it sufficiently well)—it has a decisive role in the design. A dot in itself is not dynamic but appears stable. The dot is often used in design to mark and emphasize significant areas. However, the dot also appears as a 2nd order element in the form of intersections. Especially when different lines converge, the intersection is often the point in the design where attention flows. If many dots are used together, as in a swarm of dots, the dot loses its calmness and appears dynamic, if not bustling.

Another particular form is the *cross*. Although composed of basic forms, the cross is naturally found in many different areas. For example, the cross represents the largest possible area to be taken by a human body and often stands for openness and balance, but also a stop or halt for orientation. The cross is found in crossroads and compasses but also arises again as a 3rd order shape in UIs in the intersections of various interface elements. Thus, the cross is omnipresent and helps to get a good structure into the interface and to direct the orientation and, therefore, the view of the users. As a 1st order design element for UIs, the cross is rarely used, and when it is, it tends to indicate a termination or a stop.

Lines

Just like dots and crosses, lines are also a particular design case. However, since lines can be an essential element, especially in the design of UIs, lines are given this special section.

According to Heimann and Schütz (2017), lines do not occur naturally. Lines arise in our perception at the point where two colored surfaces meet. This border can be the sky separating itself from the ground, a tree sticking out of the forest, or a river running through grass. As a design element, however, lines can be used deliberately to guide the viewer's gaze.

Lines are more dynamic than dots because they always seem to run in one direction. The only exception is the horizontal line, which tends to convey calm. This line is also the most reminiscent of the natural horizon line.

In addition to the lines' alignment, you can play with their thickness and "wholeness." Thicker lines look more stable and robust; finer lines are more graceful and less intrusive. Of course, you can also vary the thickness of a line as it progresses. *Wholeness* refers to whether you use a continuous line or a dashed line. A continuous line is firmer and more definitive; a dashed line is more porous and less solid, as shown in Figure 7.56.

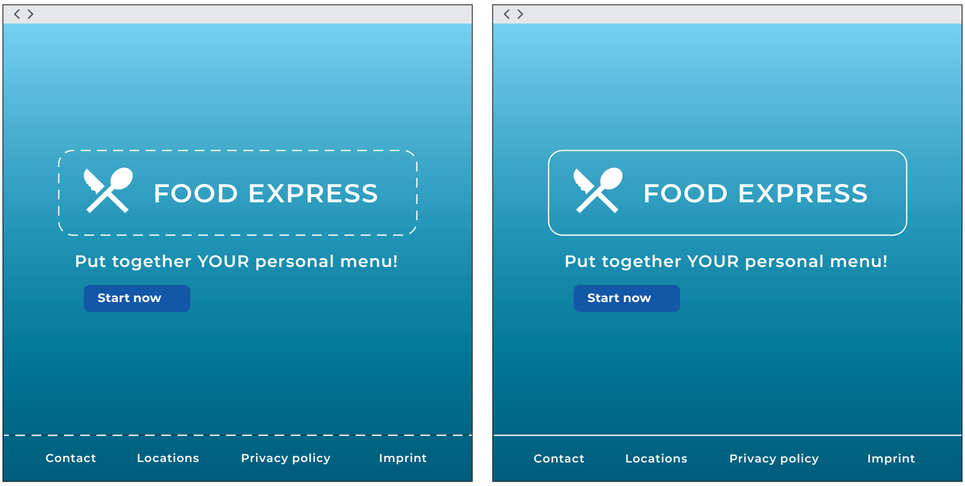

FOOD EXPRESS
Put together YOUR personal menu!
Start now
Contact
Locations
Privacy policy
Imprint
FOOD EXPRESS
Put together YOUR personal menu!
Start now
Contact
Locations
Privacy policy
Imprint

Figure 7.56 UI with dashed lines or solid lines used for delimitation purposes.

More information on lines in all their forms can be found in Heimann and Schütz (2017), which also lists various influencing factors.

Materials and Textures

For the design of physical products, the material and texture have always played a significant role. In some industries, the material is so important that a sample can be ordered before a product is purchased (for example, fabrics). However, the focus of this book is on usability and UX design, with an emphasis on digital UIs. Therefore, we'll limit ourselves to the digital representation of material and texture.

One of the main functions of using materials and textures in the digital world is to clarify the character traits of the depicted object. Through the image, specific associations are called forth in the viewers. Therefore, it is essential to consider not only the texture of an object as such but the interplay between the texture and the rest of the design.

Especially in video games, material and texture are used extensively. For example, in role-playing games, a celebrated milestone is always when representations become so detailed and realistic that you can almost feel the material. Proper rendering plays a crucial role in player immersion (i.e., the feeling of being in the story), as shown in Figure 7.57.

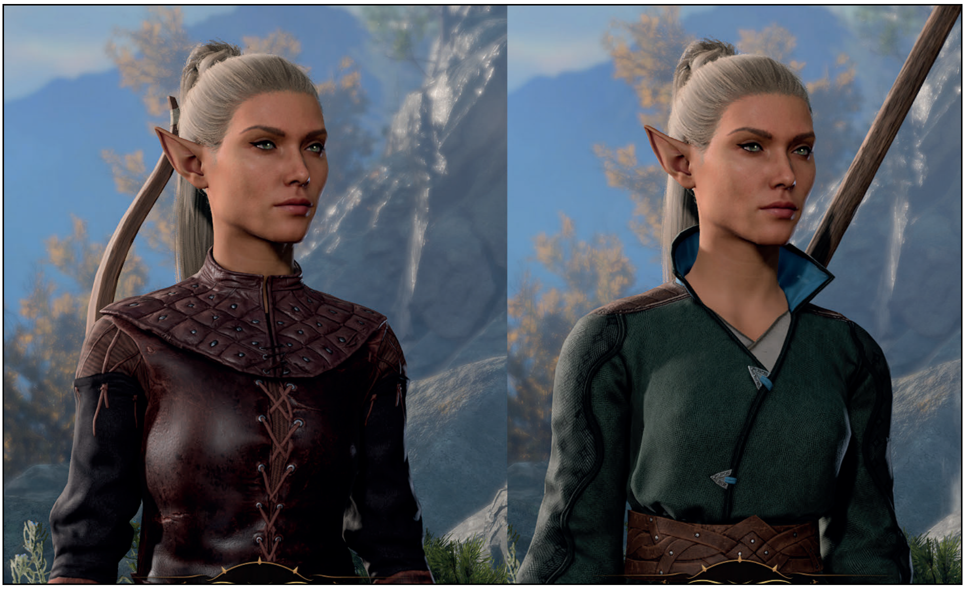
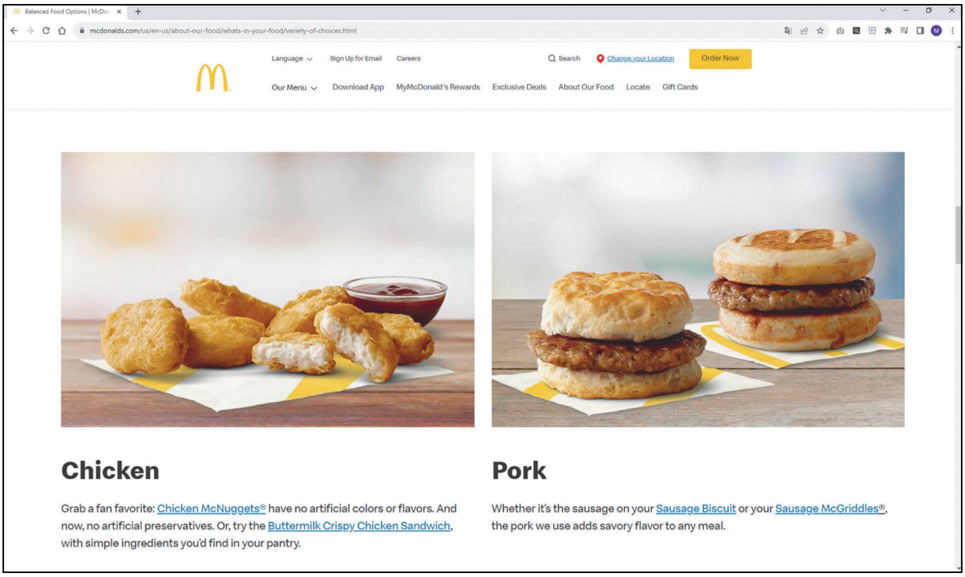


Figure 7.57 Different armor/clothing materials in the game Baldur's Gate 3.

Since the associations triggered are significantly related to a person's personal experiences, textures are also always culturally influenced. The most substantial effect is caused by a texture representing a material the users know well. Then the image also triggers a stronger connection between the induced feeling and the original use. Gold, for example, is often associated with jewelry or high value.

Using the right textures, you can thus either reinforce an already existing impression (for example, if you use wood textures when it comes to naturalness) or deliberately fight against existing associations. For instance, in the marketing of industrially produced foods on the associated product websites, textures are often deliberately used to suggest a more familiar and factory-like production, as shown in Figure 7.58.


Balanced Food Options | McDon
mcdonalds.com/us/en-us/about-our-food/whats-in-your-food/variety-of-choices.html
Language
Sign Up for Email
Careers
Search
Change your Location
Order Now
Our Menu
Download App
MyMcDonald's Rewards
Exclusive Deals
About Our Food
Locate
Gift Cards

Chicken

Grab a fan favorite: Chicken McNuggets® have no artificial colors or flavors. And now, no artificial preservatives. Or, try the Buttermilk Crispy Chicken Sandwich, with simple ingredients you'd find in your pantry.

Pork

Whether it's the sausage on your Sausage Biscuit or your Sausage McGriddles®, the pork we use adds savory flavor to any meal.

Figure 7.58 Product photo for McDonald's products. The materials used are mainly wood, paper, and glass.

Materials or textures are, therefore, well suited to triggering strong associations. On the one hand, you now have another design element to use; on the other hand, you can define a design much more tightly than you would by using colors and shapes alone because materials are associated with real objects in the user's everyday life. Therefore, these associations are more challenging to use for a general purpose. Depending on your goal, this design element can be a real advantage.

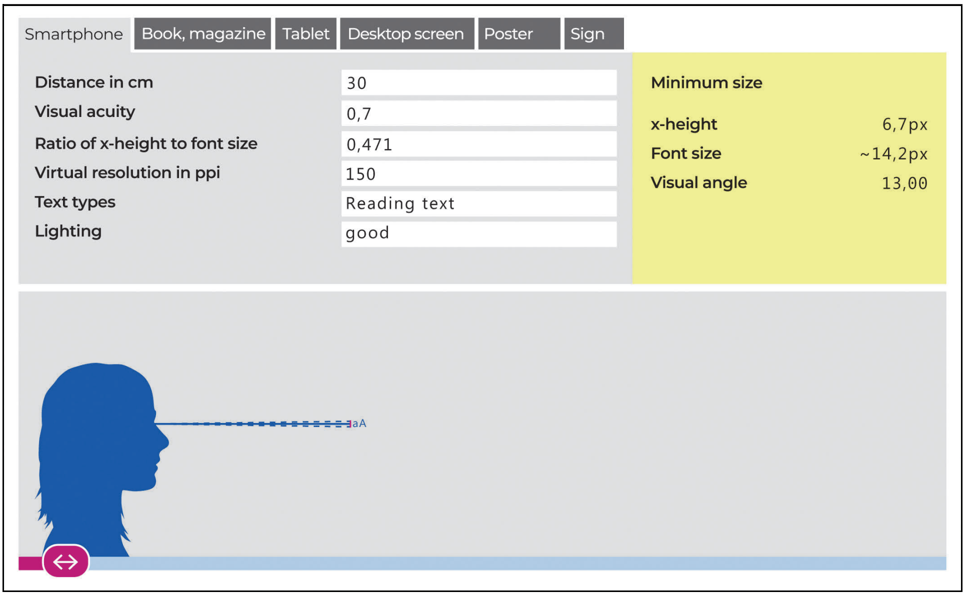
Typography

Typography is an essential topic in visual design. Nevertheless, we only want to focus on this topic briefly here and refer you instead to the book *Exploring Typography* by Tova Rabinowitz (2015), which covers the topic in almost 500 detailed pages. We cannot come close to offering this depth in this book. Therefore, we would like to focus only on a few issues of particular importance for usability and UX design.

The Legibility of Fonts

The first and—at least for usability and UX design—the most crucial topic in the field of typography is the legibility of your text. *Legibility* is about whether the individual characters can be perceived and distinguished. When you decide on a font, you should therefore think about whether this font is legible.

Regarding legibility, ISO 24509:2019 deals with the legibility of fonts. The ISO standard specifies a method for calculating the minimum font size. Alternatively, the website *www.leserlich.info*, which we introduced earlier in our discussion of contrast ratios, also has a calculator for determining the minimum font size. For this calculation, specify the screen distance, the presumed visual acuity (the viewer's visual strength), the lighting conditions, the text type, the display resolution, and the ratio of x-height to font size (shown later in Figure 7.60) to determine the minimum font size, as shown in Figure 7.59.



Distance in cm	30
Visual acuity	0,7
Ratio of x-height to font size	0,471
Virtual resolution in ppi	150
Text types	Reading text
Lighting	good

Minimum size

x-height	6,7px
Font size	~14,2px
Visual angle	13,00

Figure 7.59 A font size calculator from www.leserlich.info.

This website also considers how fonts can be displayed on different media, allowing you to switch between calculations for smartphones, tablets, and desktops or for print products.

The *corpus size* is the height of a lowercase letter in the font (for example, a lowercase "e"), and the *font size* is the length from the topmost point of the highest letter in the font to the lowest point of the lowest letter in the font, as shown in Figure 7.60.

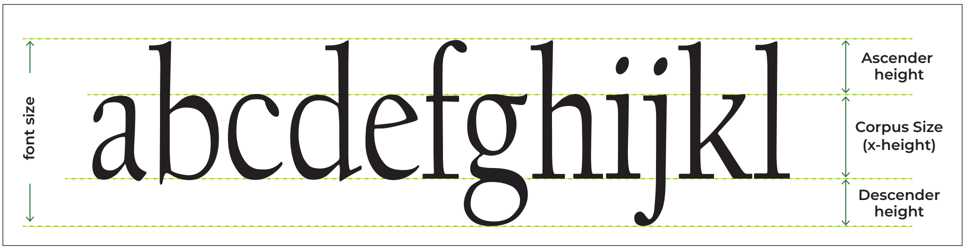


Figure 7.60 Different dimensions in typography.

Regarding legibility, in addition to font size, the *font type* is a hotly debated topic. Various ways exist to divide fonts into classes. One of the most common—and perhaps most important—divisions is into fonts with serifs and fonts without serifs. *Serifs* are the small strokes on the letters, as shown in Figure 7.60, at the top and bottom of the "b" or "i." It is widely agreed that fonts with serifs are more legible than those without serifs, especially for continuous text.

Although studies have shown this rule to be true, *sans-serif fonts* have become the preferred font type for web and digital applications, whereas *serif* typefaces are still used primarily in print. This preference is partly because sans-serif fonts are often perceived as more modern. Whether you follow this trend or opt for better legibility is, of course, up to you. In general, however, you should remember that—as with all other design elements—you influence your users' associations through font choice, as shown in Figure 7.61.

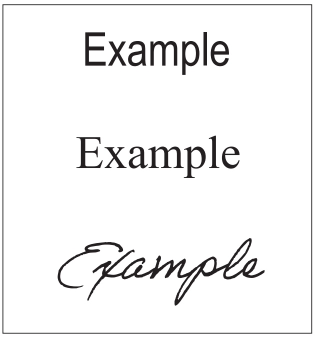


Figure 7.61 The effect of different fonts on the same word: without serifs (top), with serifs (middle), and handwritten (bottom).

In addition to serifs as a distinguishing feature, some fonts emulate the look of *handwriting*. These fonts often have a festive look and can be used well as headlines. However, text written exclusively in digital handwriting loses a lot of legibility and is not recommended.

The Amount of Text

In general, you should always consider the amount of text when designing digital applications. Unless you're offering a blog article or writing an explainer, your objective should be to produce as little text as possible. The text you have in your digital application should be primarily to guide the user.

Unlike images, texts must always be "deciphered." So, information is lost if your users can't read well or can't read at all. Images, on the other hand, can always be perceived—except by people with visual impairments. When designing UIs, you should avoid long explanatory texts as much as possible and instead use single words that can be quickly grasped. Images and text are often used together so that the image conveys the core message, and the text makes the content more concrete.

Even if not directly about typography, we would like to recommend the books *Strategic Writing for UX* by Torrey Podmajersky (2019) and *UX Writing & Microcopy* by Kinneret Yifrah (2019). In these books, you'll learn more about using text wisely to optimize usability and UX.

Images and Graphics

Images and graphics are a core component of design. However, how much space images and graphics take up in the design often depends on the objective of the system, product, or service.

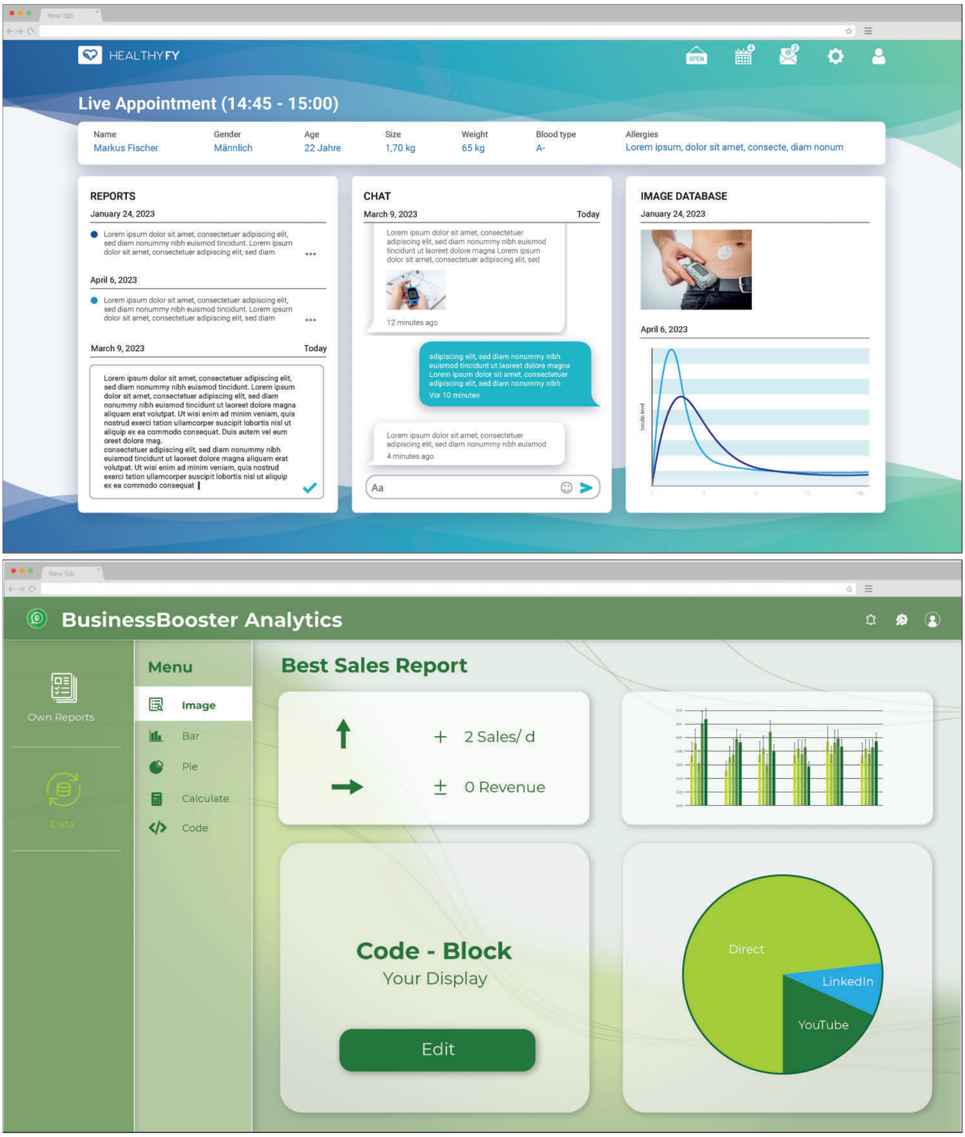
If we look at the example projects from this book, as shown in Figure 7.62, much less thought was given to graphics and images in the design of BusinessBooster than in the case of Healthyfy. In BusinessBooster, most attention was given to data input and output; thus, the focus was on input fields. In Healthyfy, on the other hand, graphics play a significant role—once on the personal level ("Who am I talking to anyway?"), but also on the content level ("What does the symptom look like?").

Regardless of the application, however, the fact is that we humans are much more designed to process visual information than other types of information. According to Heimann and Schütz (2017), about 90% of the information processed is visual, which is processed up to 60,000 times faster than text. The use of images and graphics can therefore lead to information being perceived more quickly and easily than would be the case with text.

One of the main advantages of images is that they can convey much more complex information than possible with text in the same amount of time. Images—positive or negative—always tell some story and thus trigger associations in the viewer. So, as with all other elements, you must ensure that all the parts you choose fit together and pay into the same associations.

How strong the triggered *association* is depends greatly on the selected image. The more precise the image is, the less room for association remains. Strong images with clear associations will lead to similar ideas in many people. Which images you use, however, depends—as with many things in design—also firmly on the cultural background of the person looking at them. Conversely, less strong images allow more scope for the association. One clear disadvantage is if the breadth of associations and the resulting ambiguity are not desired. To check whether the users assign the desired associations to the images, you can use the ARTTest described in Chapter 12, Section 12.19, which can determine whether the images have the desired associations.

Graphics and images, therefore, have the property of being able to convey a lot of information quickly. They also attract a lot of *attention* because they are the preferred source of information. For usability and UX design, you should use images and graphics to strengthen the desired associations. In general, however, all images and graphics should contribute to the message and content and not distract from it.



Name: Markus Fischer
Gender: Männlich
Age: 22 Jahre
Size: 1,70 kg
Weight: 65 kg
Blood type: A-
Allergies: Lorem ipsum, dolor sit amet, consecte, diam nonum

January 24, 2023

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam •••

April 6, 2023

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam •••

March 9, 2023 Today

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum oreet dolore mag.
consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat ✓

March 9, 2023 Today

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed

12 minutes ago

adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh

Vor 10 minuten

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod

4 minutes ago

January 24, 2023

April 6, 2023

+ 2 Sales/ d
± 0 Revenue

Figure 7.62 Example screens from Healthyfy and BusinessBooster with their different use of graphics and images.

Putting a strong image in the background that has no relation to the actual system, product, or service may look appealing at first. Still, a busy background reduces the usability and UX of the solution because users focus their attention on the image. The key is finding images and graphics that offer added value in content. You'll ensure that your users better absorb the content of the UI and can understand it more quickly.

Icons

A particular form of graphics is the icon, a graphical symbol primarily used in our field for application programs or files on the screen. In contrast to pictures or graphics, icons often represent a highly abstract form of content.

The advantage of icons is that they can be perceived much faster than text, similar to graphics or images. Using icons helps users find their way around the UI quickly and easily. Virtually no application can do without icons.

However, due to their high level of abstraction and usually tiny size, icons have the disadvantage of often not being independently understandable. Still, the users must first learn their meaning. This learning can occur within a system, product, or service by providing the icons with additional labeling.

In addition, however, standard icons are also frequently used. They stand for the same or similar activities across different applications. An example is the floppy disk icon for saving, even though computers have not had floppy disk drives for many years, as shown in Figure 7.63.

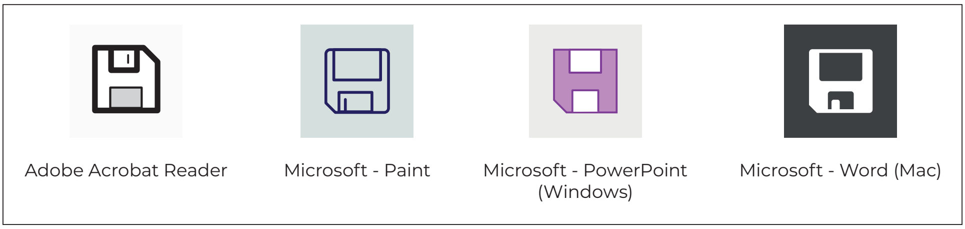


Figure 7.63 Floppy disk icons for saving from different applications.

Even if small size limits the level of detail in icons, room for design still exists. In principle, the style of the icons can vary significantly, in that the same icon is displayed once flat and two-dimensional or spatial and three-dimensional. Just as well, you can play with colors within the icon. Some icons consist exclusively of contour lines, and some are more or less colorful, as shown in Figure 7.64.

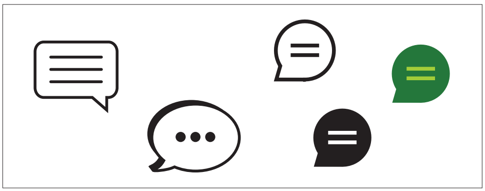


Figure 7.64 The same icon in different versions.

Our recommendation for good usability and UX is the icons should match everything else, but their level of detail should be as low as possible. When you design your icons, you should always remember that, in most cases, they are quite small on the screen afterward, and their actual size also depends on the screen resolution. The more details you integrate, the larger the icon must be later to be recognized. So, in general, the more critical the icon is and the larger it will be displayed, the more detail you can include. Regardless of the level of detail, of course, it is always true that its meaning must be understood (at least in combination with the caption), as shown in Figure 7.65.

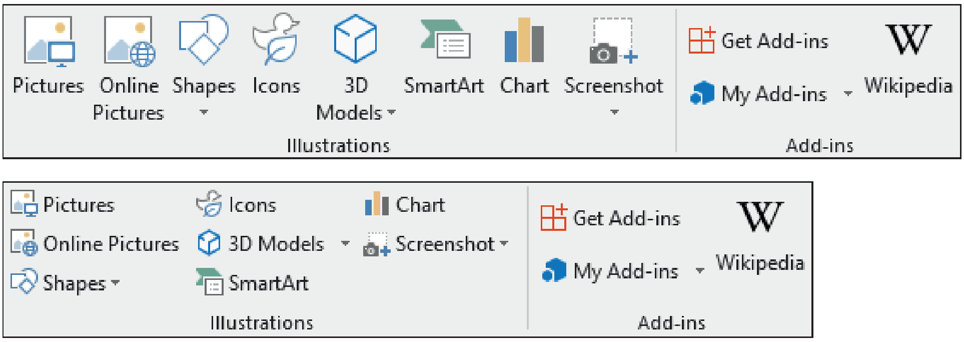

Pictures
Online Pictures
Shapes
Icons
3D Models
SmartArt
Chart
Screenshot
Illustrations
Get Add-ins
My Add-ins
Wikipedia
Add-ins
Pictures
Icons
Chart
Online Pictures
3D Models
Screenshot
Shapes
SmartArt
Illustrations
Get Add-ins
My Add-ins
Wikipedia
Add-ins

Figure 7.65 Microsoft Word's icon bar: Larger icons (top) for larger screens, and small icons (bottom) for more limited space.

7.4.3 Gestalt Principles

Up to this point, we have presented various fundamentals from psychology and design. Now, step by step, we would like to venture closer to the application-oriented basics: What happens when design and perception come together? The result of dealing with this question are *Gestalt principles*, which provide simple laws to understand the perception of design better and thus make better design decisions. These principles go back to Max Wertheimer (1923).

In design, unfortunately, not only must every single element look good. You also must think about how the different components work together. Let's start with the law of proximity.

Law of Proximity

The *law of proximity* states that elements in spatial proximity are perceived as belonging. For example, to perceive a label as belonging to a specific field, it must be placed in spatial proximity to the field. The closer elements are to each other, the easier the assignment usually is, as shown in Figure 7.66.

On the left, deciding which field the label belongs to is challenging unless you look at the entire interface from top to bottom. In this version, the field label could also have been placed below the field. This relationship becomes apparent only by spatially placing the label near an element. You can see how this placement affects perception and interpretation differently on the right, where there is less room for interpretation.

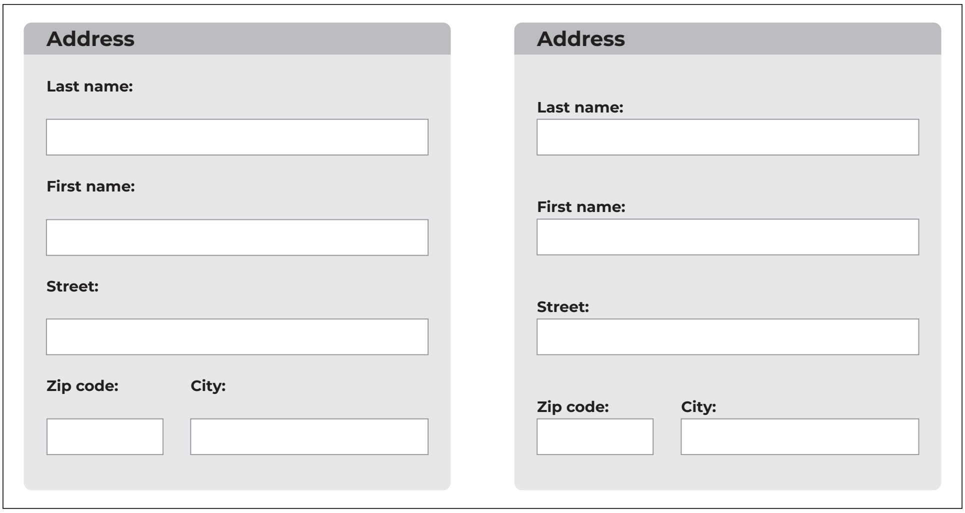

Address
Last name:
First name:
Street:
Zip code:
City:
Address
Last name:
First name:
Street:
Zip code:
City:

Figure 7.66 The effect of spatial proximity on the affiliation of individual elements to each other.

Law of Similarity

Another law is the *law of similarity*, which states that similar elements are perceived as belonging together. Thus, when designing UIs, attention must be paid to the visual separation of similar but functionally different elements.

A typical mistake in the design of forms, for example, is to design text fields where users are supposed to make entries that look exactly like text fields where explanatory texts are displayed by the system, as shown in Figure 7.67.

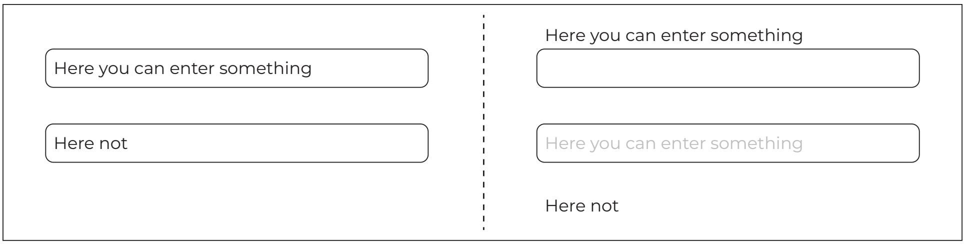

Here you can enter something
Here not
Here you can enter something
Here you can enter something
Here not

Figure 7.67 Poor design that does not consider the law of similarity (left) and a better solution (right).

This mistake can even happen when the content has little to nothing to do with each other. Imagine, for example, that you're at an engine test bench monitoring engine speed and fuel level in software. In general, the displays could look quite similar visually (semicircle scale), except that in one case, a high value is good, and in the other case, a low value. If the design is poor, the displays would be mixed and sorted only by vehicle. In a better design, the displays would be sorted by vehicle and type, as shown in Figure 7.68.

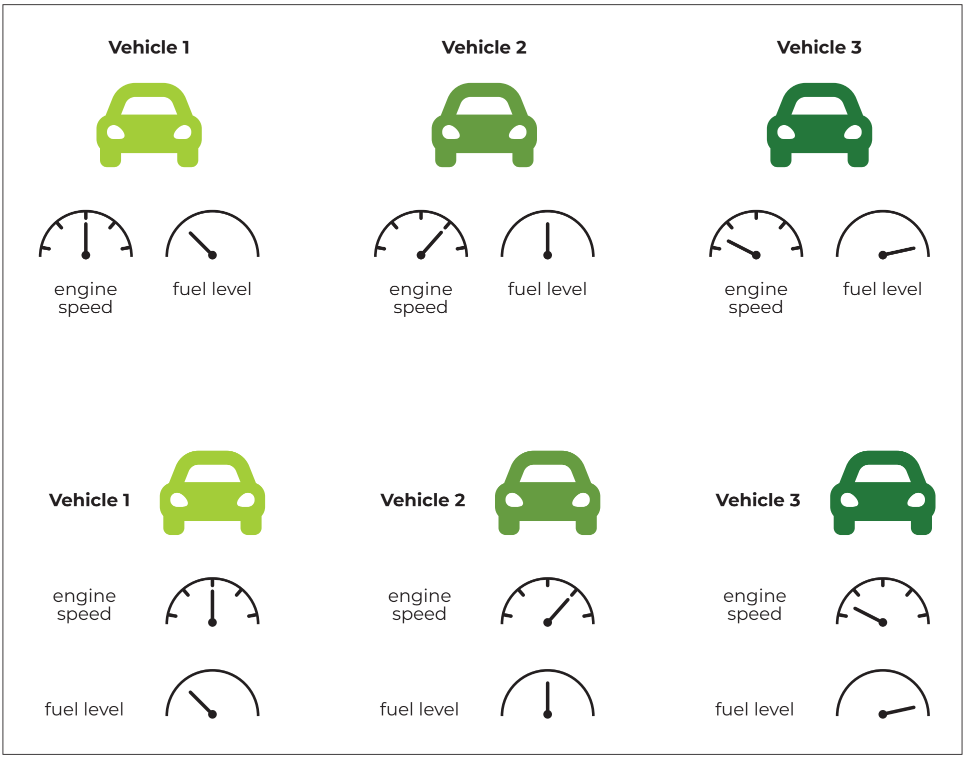

Vehicle 1
Vehicle 2
Vehicle 3
engine speed
fuel level
engine speed
fuel level
engine speed
fuel level
Vehicle 1
Vehicle 2
Vehicle 3
engine speed
engine speed
engine speed
fuel level
fuel level
fuel level

Figure 7.68 Unfavorable (top) and favorable (bottom) use of the law of similarity to enhance UI understanding.

It's essential for you to know that the law of similarity refers to various aspects. Thus, elements can be perceived as similar with:

- The same color
- The same texture
- The same alignment
- The same style
- The same direction of motion
- The same size
- The same speed

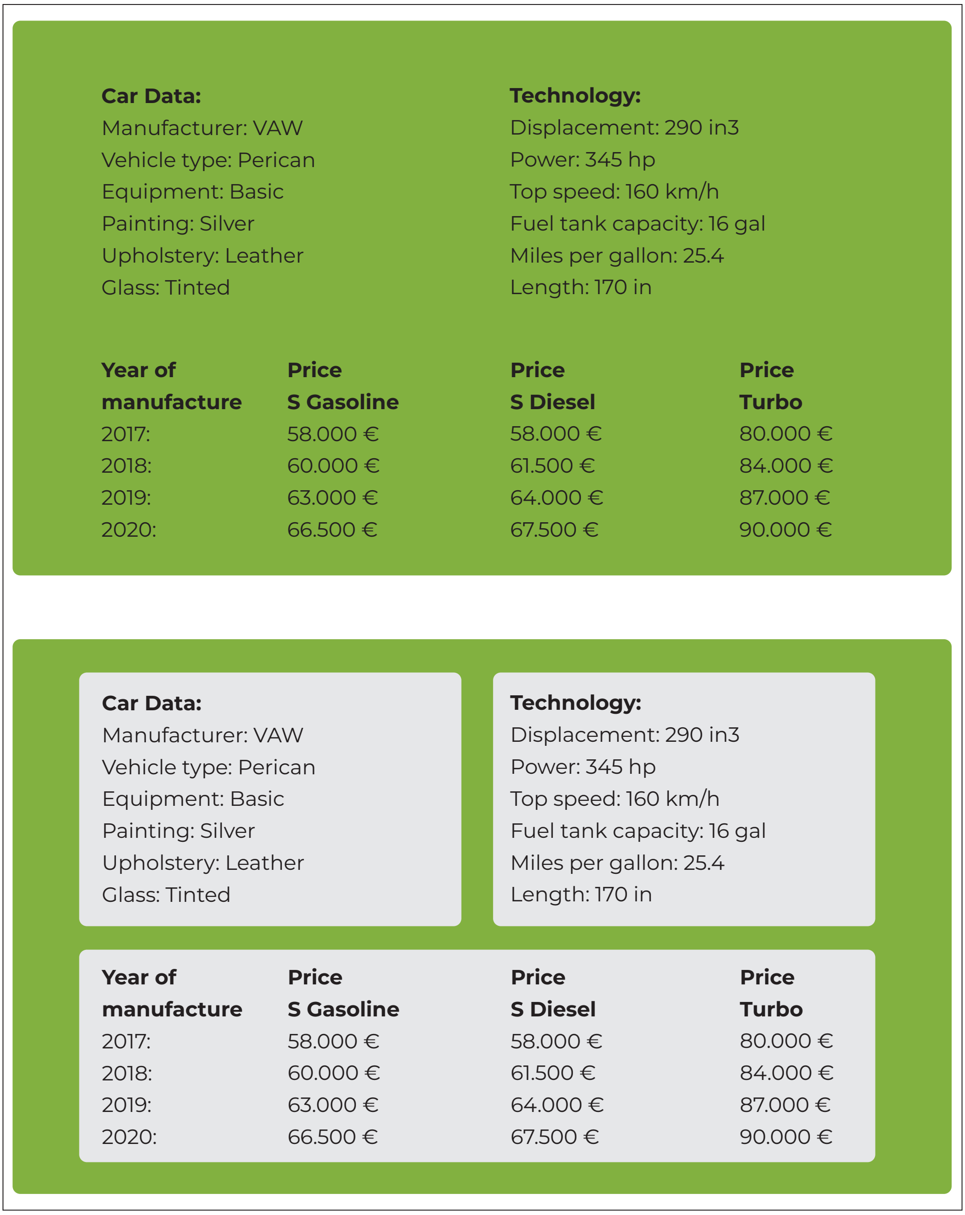
Thus, you can use various design aspects to ensure that the proper elements are perceived as belonging together.

Law of Closure

Another Gestalt principle is the *law of closure*, which states that the viewer fills in missing components of a "closed" design. Applied to UIs, users often look for a structure and a higher-level logic to move more quickly through the interface.

However, elements in UIs often fail to create a “complementary” shape, making grouping challenging to perceive and making it harder to find information. You can help your users find the structure by designing your UI to highlight them visually.

For the example shown in Figure 7.69, whether each text block forms a unit by itself is unclear.



Car Data:
Manufacturer: VAW
Vehicle type: Perican
Equipment: Basic
Painting: Silver
Upholstery: Leather
Glass: Tinted

Technology:
Displacement: 290 in3
Power: 345 hp
Top speed: 160 km/h
Fuel tank capacity: 16 gal
Miles per gallon: 25.4
Length: 170 in

Year of manufacture	**Price S Gasoline**	**Price S Diesel**	**Price Turbo**
2017:	58.000 €	58.000 €	80.000 €
2018:	60.000 €	61.500 €	84.000 €
2019:	63.000 €	64.000 €	87.000 €
2020:	66.500 €	67.500 €	90.000 €

Car Data:
Manufacturer: VAW
Vehicle type: Perican
Equipment: Basic
Painting: Silver
Upholstery: Leather
Glass: Tinted

Technology:
Displacement: 290 in3
Power: 345 hp
Top speed: 160 km/h
Fuel tank capacity: 16 gal
Miles per gallon: 25.4
Length: 170 in

Year of manufacture	**Price S Gasoline**	**Price S Diesel**	**Price Turbo**
2017:	58.000 €	58.000 €	80.000 €
2018:	60.000 €	61.500 €	84.000 €
2019:	63.000 €	64.000 €	87.000 €
2020:	66.500 €	67.500 €	90.000 €

Figure 7.69 A UI lacking “closed” design elements (top) and a UI supporting users through closure (bottom).

Visually, the blocks “Car data,” “Year of manufacture,” and “Price S Gasoline” as well as “Technology,” “Price S Diesel,” and “Price Turbo” could also belong together. In the lower part, however, the structure becomes recognizable.

Law of Continuity

The *law of continuity* states that elements are perceived as belonging together when they appear as a continuation of previous elements.

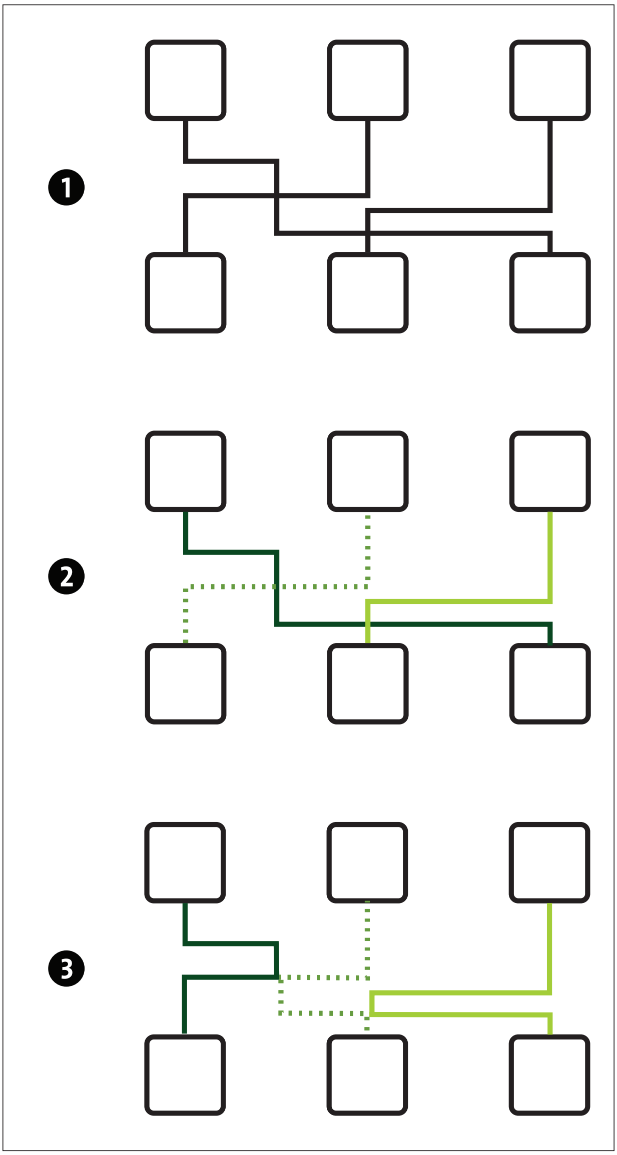


Figure 7.70 How the law of continuity influences the perception of individual representations.

Again, users try to discover a logic in the design. Looking at ❶ of the illustration shown in Figure 7.70, you can probably tell quite accurately from the illustration where you think the lines lead.

Presumably, you would make the assignment ❷. We have shown you an alternative ❸, which theoretically would also work. However, due to the law of continuity, the lines are more likely to be perceived as continuous.

Law of Symmetry

Another Gestalt law is the *law of symmetry*. An element is perceived as symmetrical (balanced) when its attributes are in equilibrium. Sometimes, UIs are often designed so one-sided that this perception of symmetry is disturbed. Especially when using forms, a one-sided alignment is usually found in the UI. You can avoid this lopsidedness by ensuring the elements are evenly distributed, as shown in Figure 7.71.

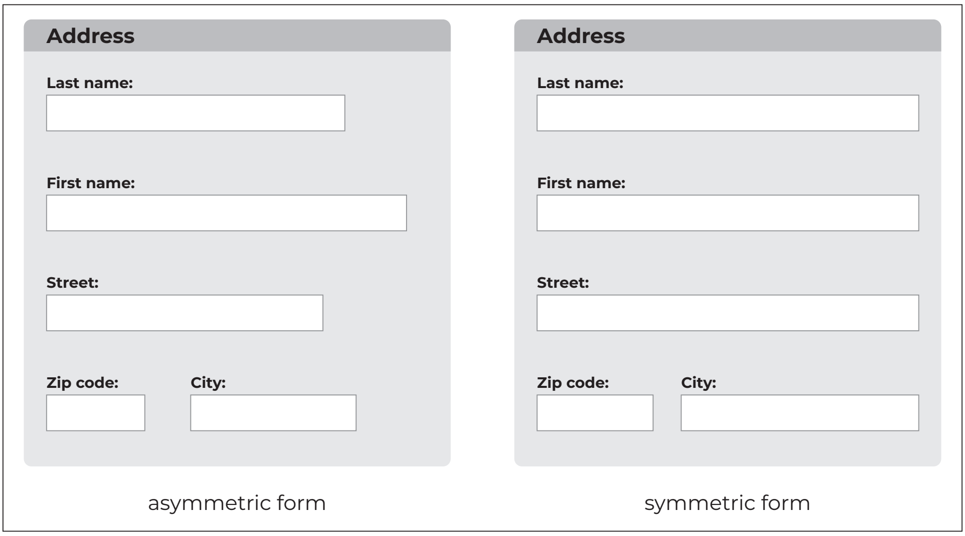
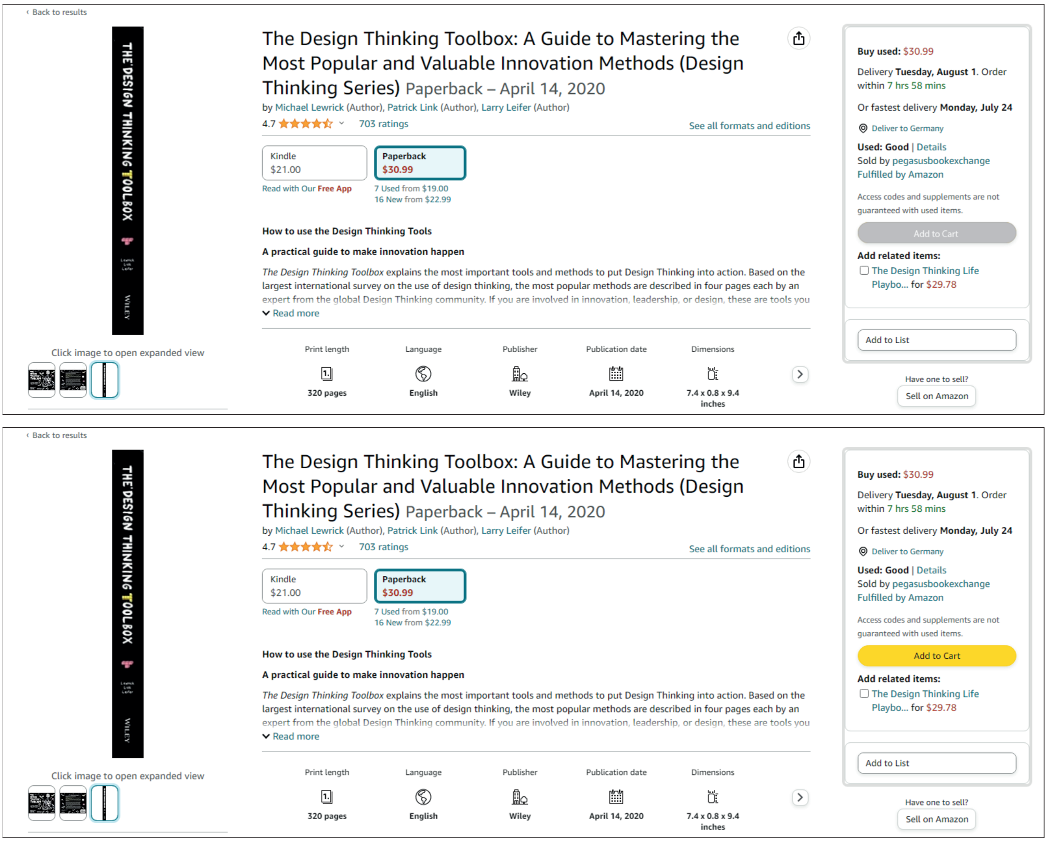


Figure 7.71 Examples of an asymmetric (left) and a symmetric (right) form.

Law of the Point of View

A law of design often used unconsciously is the *law of the point of view*. When you use one element significantly different from the rest, you draw attention to this element. This approach is regularly used in website design to make the CTA stand out clearly from the rest. This choice lets you control the user's gaze with simple means, as shown in Figure 7.72.



The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods (Design Thinking Series) Paperback – April 14, 2020

by Michael Lewrick (Author), Patrick Link (Author), Larry Leifer (Author)

4.7 703 ratings

How to use the Design Thinking Tools

A practical guide to make innovation happen

The Design Thinking Toolbox explains the most important tools and methods to put Design Thinking into action. Based on the largest international survey on the use of design thinking, the most popular methods are described in four pages each by an expert from the global Design Thinking community. If you are involved in innovation, leadership, or design, these are tools you

Print length: 320 pages
Language: English
Publisher: Wiley
Publication date: April 14, 2020
Dimensions: 7.4 x 0.8 x 9.4 inches

Buy used: $30.99

Delivery **Tuesday, August 1**. Order within 7 hrs 58 mins

Or fastest delivery **Monday, July 24**

Deliver to Germany

Used: Good | Details

Sold by pegasusbookexchange

Fulfilled by Amazon

Access codes and supplements are not guaranteed with used items.

Add related items:

☐ The Design Thinking Life Playbo... for $29.78

The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods (Design Thinking Series) Paperback – April 14, 2020

by Michael Lewrick (Author), Patrick Link (Author), Larry Leifer (Author)

4.7 703 ratings

How to use the Design Thinking Tools

A practical guide to make innovation happen

The Design Thinking Toolbox explains the most important tools and methods to put Design Thinking into action. Based on the largest international survey on the use of design thinking, the most popular methods are described in four pages each by an expert from the global Design Thinking community. If you are involved in innovation, leadership, or design, these are tools you

Print length: 320 pages
Language: English
Publisher: Wiley
Publication date: April 14, 2020
Dimensions: 7.4 x 0.8 x 9.4 inches

Buy used: $30.99

Delivery **Tuesday, August 1**. Order within 7 hrs 58 mins

Or fastest delivery **Monday, July 24**

Deliver to Germany

Used: Good | Details

Sold by pegasusbookexchange

Fulfilled by Amazon

Access codes and supplements are not guaranteed with used items.

Add related items:

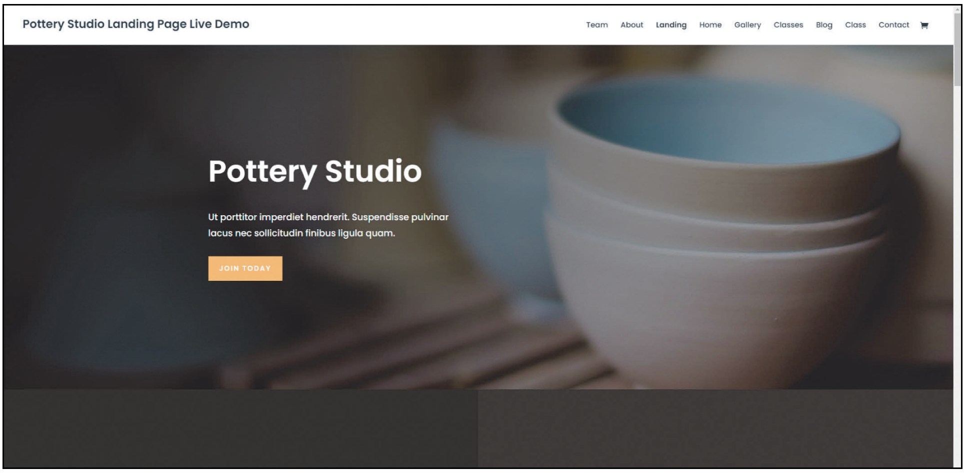
☐ The Design Thinking Life Playbo... for $29.78

Figure 7.72 A UI that does not use the law of point of view (top) and a UI that does (Amazon.com).

Law of Figure/Ground

In the design of UIs, the *law of figure/ground* plays a significant role. This law questions how to declare which elements users need to focus on for interaction and which elements are only used as decoration.

To help users make these distinctions as quickly as possible, make sure that the elements in focus have a clear shape and boundary, whereas the background is somewhat amorphous. The background should also continue without gaps behind the elements in focus so that the elements obscure parts of it. All elements in focus should appear closer and at a fixed position, whereas the background may appear farther away and positionless, as shown in Figure 7.73.



Pottery Studio Landing Page Live Demo

Pottery Studio

Ut porttitor imperdiet hendrerit. Suspendisse pulvinar lacus nec sollicitudin finibus ligula quam.

Figure 7.73 A unique figure/ground display shown on the homepage at https://www.elegantthemes.com/layouts/art-design/pottery-studio-landing-page/live-demo. The text and the button are clearly visible and in front of the blurrier background photo.

Law of Common Fate

The *law of common fate* has become increasingly important for the design of UIs, especially in recent years. This law states that elements are perceived as belonging together if they change simultaneously or move similarly.

Recently, more and more websites and software solutions have incorporated animations that move individual elements or areas of the UI. If you plan to do the same, make sure that the elements that belong together also move together. Since animation is challenging to show in a book, we'll try to clarify this law using suggested directions of movement, as shown in Figure 7.74.

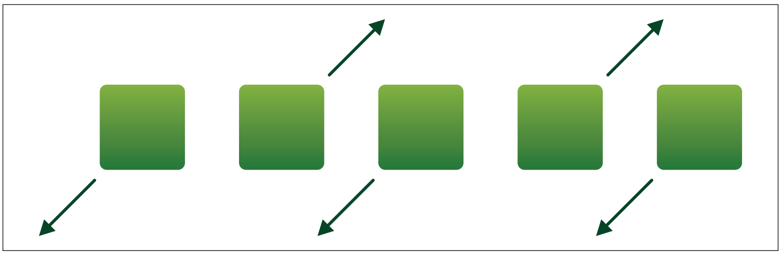


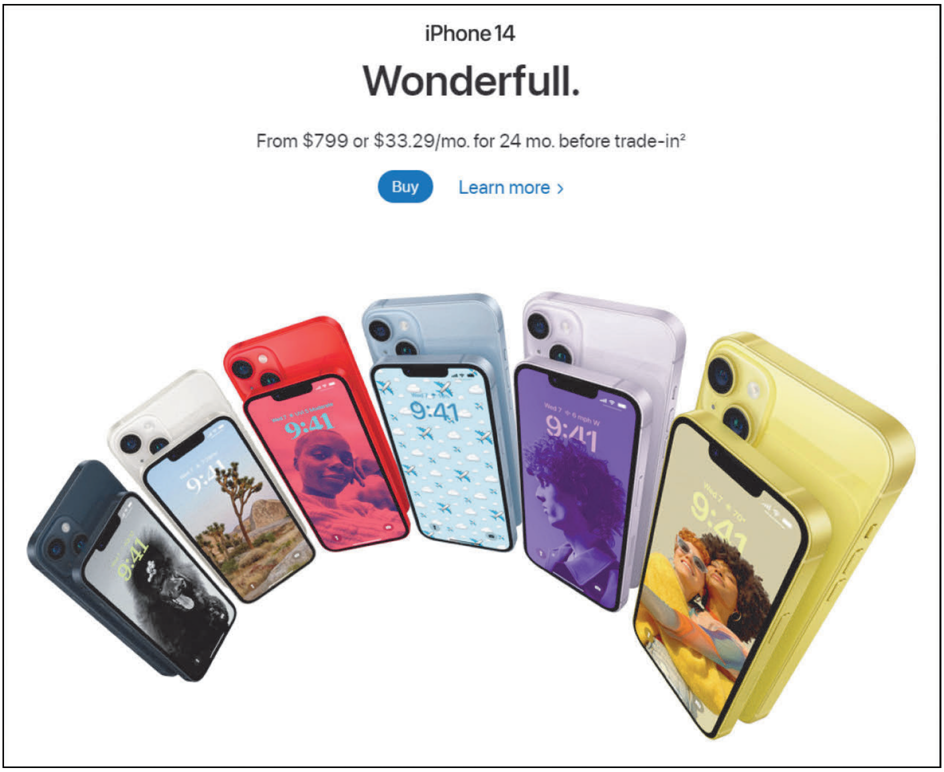
Figure 7.74 The law of common fate.

Due to the law of common fate, shapes moving in the same direction are perceived as belonging to each other. Viewed from left to right, the first, third, and fifth elements move downward to the left according to the arrow. Thus, they form a group. In turn, the second and fourth elements move upwards to the right and thus create a second group.

Law of Visual Hierarchy

Whether a UI is later deemed well structured and easy to understand depends heavily on whether it succeeds in presenting a clear order. The *law of visual hierarchy* helps you to create this order precisely. You can influence the perceived order by using an element's particular size, special shape, or other attributes, as shown in Figure 7.75.

This law is in play in this book because the main chapters have larger headings than subchapters and other sections. Usually, the essential information is visually emphasized by being more prominent and appearing “stronger” than the rest.



iPhone 14

Wonderfull.

From $799 or $33.29/mo. for 24 mo. before trade-in[2]

Figure 7.75 Example of the law of visual hierarchy from https://www.apple.com/iphone/.

In the text, the headline “Wonderfull.” has the highest hierarchy level, followed by “iPhone 14” and the pricing text and button. Overall, however, the image is also very dominant.

The Gestalt Principles in a Nutshell

In summary:

- **Law of proximity**
  Elements close to each other are perceived as belonging together.

- **Law of similarity**
  Visually similar elements are perceived as belonging together.
- **Law of closure**
  The observer completes missing components of a "closed" shape.
- **Law of continuity**
  If elements appear to continue previous elements, they are perceived as belonging together.
- **Law of symmetry**
  If the attributes of an element are in balance, then the element is perceived as symmetrical.
- **Law of the point of view**
  If one element significantly differs from the others, it attracts attention.
- **Law of figure/ground**
  Elements in the foreground have a shape. Elements in the background are somewhat formless.
- **Law of common fate**
  Elements that change or move together are perceived as belonging.
- **Law of visual hierarchy**
  The size and shape of objects influence the visual importance of the elements.

7.4.4 Interaction Principles

A handbook on usability and UX design is impossible to write without referring to ISO 9241-110, "Ergonomics of human-system interaction—Part 110: Interaction principles." This ISO standard presents seven fundamental *interaction principles* that you can apply to ensure that your system, product, or service is easy to use.

As with any standard, the principles are relatively general, so looking at the standard without many examples is of limited help. In this section, we present all seven interaction principles as clearly and as simply as possible. We'll also explore many examples that are intended to illustrate these principles.

Note that cases always exist where things will behave differently than in the examples. In the usability and UX world, the context of use is essential, and if it changes, the associated solutions also change. You may keep coming up with examples that are contrary to ours. The reason for this difference can almost always be traced back to the context of use. Therefore, do not take our examples as fixed rules but use them to understand basic principles and transfer this knowledge to your area. One good thing about these principles is that you can apply them to almost any interaction, thus allowing you to test repeatedly and ensure good usability and UX.

The interaction principles are pretty old and based on other design principles, but they are still highly relevant and consistent today. Six of the seven principles have been included in the standard for a long time, whereas the principle of user engagement replaced "suitability for individualization" only a few years ago.

Before we start presenting all the interaction principles individually, we would like to point out that the principles are always listed "next to each other" in the standard. Thus, determining which interaction principle is the most important or should receive the most attention is impossible without looking at the specific case. Individual interaction principles may be more critical than others, depending on what kind of system, product, or service you're designing. Therefore, you must decide which apply best to your case. This consideration is essential because interaction principles can sometimes contradict each other, and an improvement according to one principle can lead to a deterioration according to another principle. But now, let's get started!

Suitability for the User's Task

Even though we just said no order exists, we want to start with the interaction principle of *suitability for the user's task*, sometimes called *task adequacy*. In our perception, *suitability for the user's task* is actually one of the essential principles, if not the most important principle.

But what does *suitability for the user's task* mean? In the ISO standard, task adequacy is defined as follows:

> *"An interactive system is suitable for the user's tasks when it supports the users in the completion of their tasks, i.e., when the operating functions and the user-system interactions are based on the task characteristics (rather than the technology chosen to perform the task)." (ISO 9241-110:2020)*

Suitability for the user's task is, therefore, about providing users with the best possible support in completing their tasks. To achieve this goal, you must precisely understand users' tasks in your system, product, or service and define the processes in your solution to perform these tasks. Ensuring the tasks are relevant to the user group is also essential. To optimize the task adequacy of your solution, then according to the standard, you should ask yourself the following questions in particular:

- Is the solution even suitable for solving a specific task?
- How much effort is needed to accomplish this task?
- Are defaults available to help users successfully perform the task?

For users to decide whether the solution suits the task, the system must provide information about what it does and the expected result. If this information is not given, users are often frustrated.

For example, in Germany, we have a habit of first asking whether restaurant or café can take a credit card before entering. (Cash has long been the preferred method of payment in Germany.) This question has arisen from the lack of transparency in payment options, which can result in the embarrassing situation of needing to search for an

ATM. You're probably familiar with similar problems while shopping online, when it turns out, only at the end, that the store does not accept your preferred payment option, only payment methods you do not (want to) use. Thus, an essential goal is to communicate clearly which tasks can be performed under which conditions.

To minimize a user's effort, a task-suitable system should provide users with enough information for each step of the interaction to complete the task quickly and efficiently.

Additionally, when designing, try to automate as much as possible any steps that are only necessary because of the technology used but do not add value to the user and the task.

A final way to keep the effort as low as possible is to hide unnecessary information that does not contribute to the current task as much as possible. This editing helps users focus on one task instead of jumping back and forth between different tasks and topics.

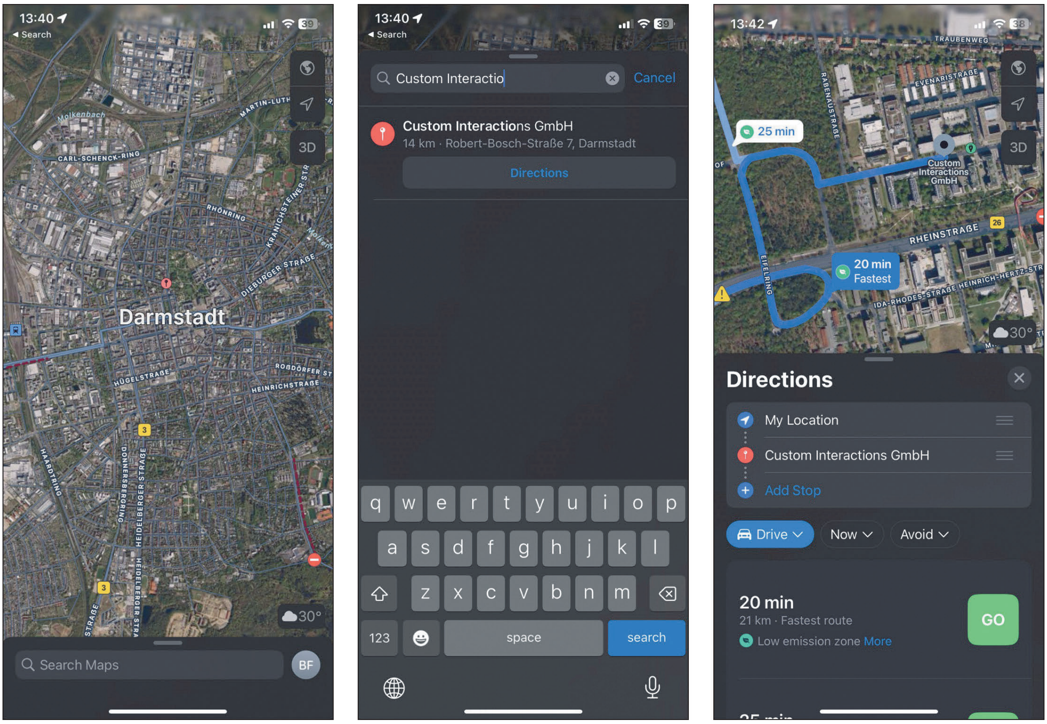

13:40
Search
3D
Darmstadt
30°
Search Maps
BF
13:40
Search
Custom Interactio
Cancel
Custom Interactions GmbH
14 km · Robert-Bosch-Straße 7, Darmstadt
Directions
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
123
space
search
13:42
25 min
3D
Custom Interactions GmbH
20 min
Fastest
30°
Directions
My Location
Custom Interactions GmbH
Add Stop
Drive
Now
Avoid
20 min
21 km · Fastest route
Low emission zone More
GO

Figure 7.76 Example: the "Maps" app on an iPhone. Left: The app has been opened. Center: The destination has been entered. Right: Navigation can be started with "Go." The most likely options are preselected (from current location, start now, travel by car).

Using automatically set defaults is a fantastic way to ensure a task is completed as easily, correctly, and quickly as possible, especially when dealing with inexperienced users or recurring tasks. In this way, you can increase task appropriateness significantly.

You’re probably familiar with a whole range of intelligent presets from your everyday life.

An example of a useful preset is the option to select your current location as the starting point in mobile navigation, as shown in Figure 7.76. Usually, you don’t need to specify where “here” is, only where you want to go. This preset usually saves a lot of time, and you only need to change this starting point in rare cases, for example, if you’re going to plan a route.

If you consider all these aspects, you can ensure that your system, product, or service fits the task and the requirements that arise from it. Put simply, you should design your solution to display as little information as possible but as much as is necessary to perform the task correctly.

Self-Descriptiveness

The second interaction principle is the self-descriptiveness:

> *“The interactive system presents appropriate information, where needed by the user, to make its capabilities and use immediately obvious to the user without unnecessary user-system interactions.” (ISO 9241-110:2020)*

A system, product, or service should explain the functions and their use to the users, which also includes using an instruction manual, so that inquiring with support or other experts is considered unnecessary.

An important point to note: Of course, some extraordinarily complex systems may not require training and familiarization. However, the necessity should always arise from the complexity of the task and not from the complexity of the UI. For example, to use a conventional navigation system to plan a route, it would be a bad sign for the system’s self-descriptiveness if you needed to read an instruction manual.

According to the standard, two aspects in particular are crucial for optimizing self-description capability:

- Is all the necessary information present and obviously available?
- Is it clear to users what stage the process is at or what state the solution is in?

If you’re thinking about the information you need, you should check whether the information is provided to guide your users through the process. Thus, in this way, no further assistance such as support, FAQs, videos, or the like becomes necessary (or at least only to a minimal extent). Again, let’s consider an example from online shopping to show how easily self-descriptiveness can be implemented, as shown in Figure 7.77.

To help your users navigate around a system, product, or service, you should also make it obvious where the user is at any given moment. Various approaches to achieving this goal exists: With websites, for example, *breadcrumbs* are often used to show where the user is currently located in the navigation, as shown in Figure 7.78.

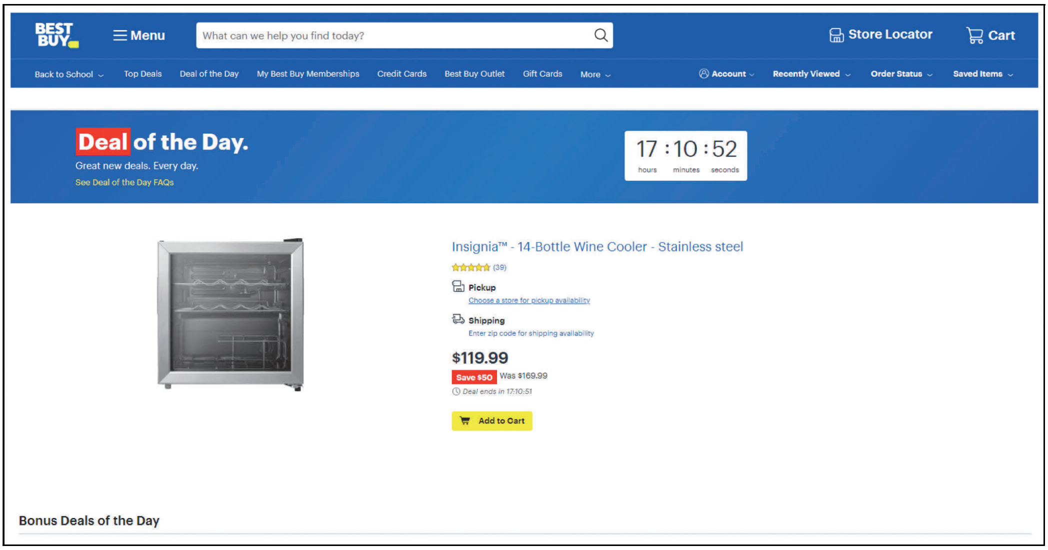

BEST BUY
Menu
What can we help you find today?
Store Locator
Cart
Back to School
Top Deals
Deal of the Day
My Best Buy Memberships
Credit Cards
Best Buy Outlet
Gift Cards
More
Account
Recently Viewed
Order Status
Saved Items
Deal of the Day.
Great new deals. Every day.
See Deal of the Day FAQs
17 : 10 : 52
hours minutes seconds
Insignia™ - 14-Bottle Wine Cooler - Stainless steel
(39)
Pickup
Choose a store for pickup availability
Shipping
Enter zip code for shipping availability
$119.99
Save $50
Was $169.99
Deal ends in 17:10:51
Add to Cart
Bonus Deals of the Day

Figure 7.77 Good self-descriptiveness in online shopping from www.bestbuy.com: The "Add to Cart" label clarifies what happens when the button is clicked and before the user does so.

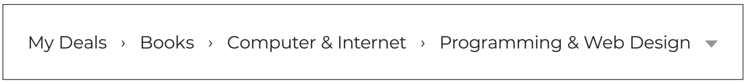

My Deals › Books › Computer & Internet › Programming & Web Design

Figure 7.78 Breadcrumbs show where you're currently located.

In other contexts, for example, the area of the menu in which the user is currently located is highlighted. No matter which solution you choose, you must support users in finding their way around and, at the same time, provide them with the necessary *controls to navigate* through the solution.

Another rule that helps increase the self-descriptiveness of a solution is to permanently show the controls used most often or necessary for the task. Even if this recommendation seems trivial at first glance, we've seen time and again important controls that can't be used because they are hidden or invisible.

Self-descriptiveness also includes marking controls not currently needed or unusable as inactive. This de-emphasis is usually achieved by graying the elements but not removing them from the display. However, if certain user groups should never access a feature (e.g., because they are not allowed to), you can safely remove it from the interface to reduce its complexity. This removal can also increase self-descriptiveness, as permanently disabled but still visible controls can distract users. (In usability and UX tests, users would repeatedly wonder how to activate this control.)

As a final note on optimizing existing information, you should check and ensure that the words you use in your UI match the language of your user group. What this means specifically for the *choice of words* depends on your user group.

When dealing with an infrequently used public application, you might want to think about commonly understood words. But things can be pretty different if you're in a work context. Especially in the case of internal applications, such as accounting, internal vocabulary often comes into play. This internal vocabulary may have to be learned first, but it makes it easier and faster for users to use. This speed may not be possible for outsiders, but you'll need to ask yourself who is the more critical user group in this case.

Another important rule is that, if possible, you should report to the users how far the solution has progressed in *completing the task*. This completion level is usually represented by numbering the steps; it would be difficult to estimate the time, since the users contribute their share of the time needed for the task. If not the case, you can also specify a time. However, most users are familiar with the display of steps from various areas, whether digital or non-digital, as shown in Figure 7.79. Also, the indication of the page number in a book is nothing more than a progress indicator.

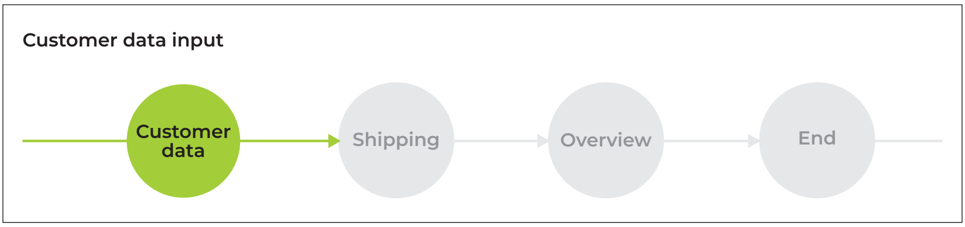


Figure 7.79 Displaying the current status of a process using a progress bar.

When something changes in the system status, you should ensure that your system, product, or service informs its users about it. The changes in system status usually impact how the system works and what is expected from users. For example, suppose you've tried to use the *Siri voice assistant* to play a particular piece of music on a highway with poor mobile internet reception. In that case, you know how important it is to inform users that the system state has changed, in this case, from "available" to "unavailable."

On the other hand, changing a system state or a new process step can also mean that the solution requires input from the user before it continues. In this case, then it should be pretty clear from the display. An excellent example is self-service checkout kiosks in shopping stores, which (if designed well) make it clear to users at all times that they are now required to scan the goods themselves or that the payment process is pending, as shown in Figure 7.80.

If you follow all of these rules and implement them in your system, product, or service, then you can be sure that users can find their way around on their own and know the current status of the interaction at all times.

Figure 7.80 Payment process during self-service checkout.

Conformity with User Expectations

Conformity with user expectations is extremely powerful because it helps you design an easy-to-use UI and helps users find their way around the solution quickly and easily. But what is conformity with user expectations anyway? The standard defines this concept in the following way:

> *"The interactive system's behavior is predictable based on the context of use and commonly accepted conventions in this context." (ISO 9241-110:2020)*

The goal of conformity with user expectations is that users find their way around directly, or at least very quickly because they can transfer what they learned previously to your system, product, or service. To avoid surprises, you should look at three different areas, according to the standard:

- Does the system behave appropriately, and does it also provide appropriate feedback to the users?
- Is the system consistent internally but also externally?
- Are there any changes in the context of use that would need to be considered?

To ensure that your system, product, or service behaves *appropriately*, you should first look at the steps needed to accomplish the task. Ensuring that the steps correspond to the steps from the user's perspective is essential. This correspondence can become especially challenging if your solution is used for many different user groups. But even then, you should try to design the system, product, or service to meet the expectations of as many people as possible.

Especially when users rarely use a system, product, or service, or are even doing so for the first time, much of the success of the interaction is related to the feedback that the system provides. In this context, you should provide user feedback as soon as possible after they have completed an interaction. A simple example is the send confirmation after a sent mail or a completed form and the confirmation mail or the confirmation screen after an online purchase.

An essential goal is that the feedback is available and understood by users. The feedback must therefore be appropriate and helpful concerning the knowledge level of your users and not about the knowledge level of your developers. Especially when error messages arise, a mismatch often becomes apparent. A good example is a modern washing machine, which often use cryptic error codes but otherwise give no indication of what the problem might be, as shown in Figure 7.81.

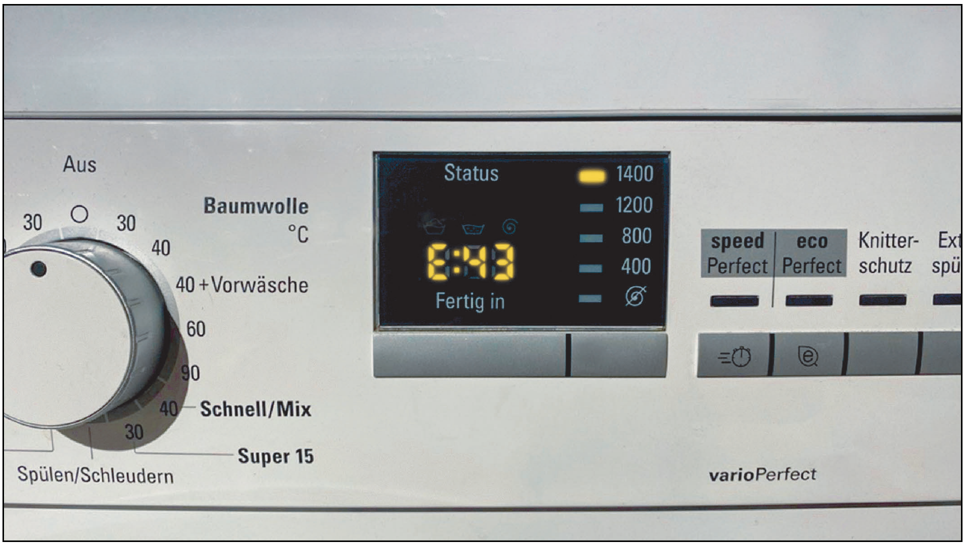


Figure 7.81 An unhelpful error message from a household appliance.

In the previous section on self-descriptiveness, we mentioned how you should inform your users about the system's status. A similar rule also applies to conformity with user expectations. However, in this context, you mainly should inform the users about the system status if an action takes noticeably longer than expected. For example, you might provide a waiting time display at a bus stop telling users that the next bus will arrive in 7 minutes. Another example is the download progress indicator for the game platform you use, which might tell you that you still need about 1 minute to download the remaining data and install a game patch, as shown in Figure 7.82.

The second area—*internal and external consistency*—is critical to help users learn the solution quickly and simplify switching between different solutions within a product family. *Internal consistency* describes whether the same display and operating elements and processes are used within the solution for the same functions. *External consistency* describes whether this also applies beyond your solution.

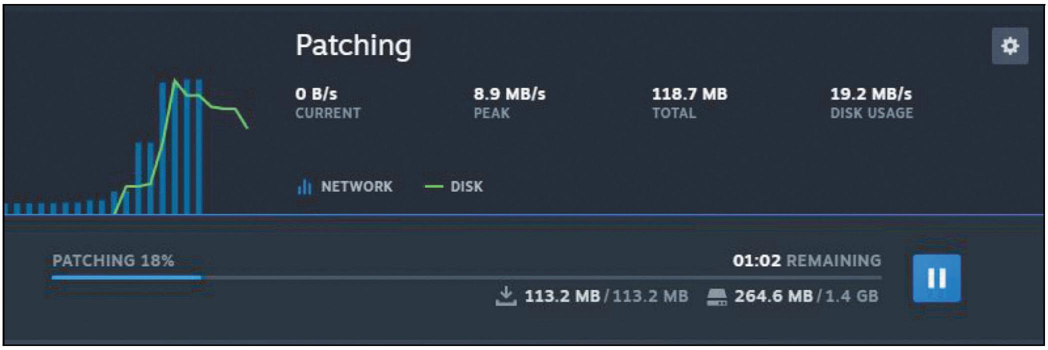


Figure 7.82 Displaying a detailed loading time clearly to users of a game client from Steam.

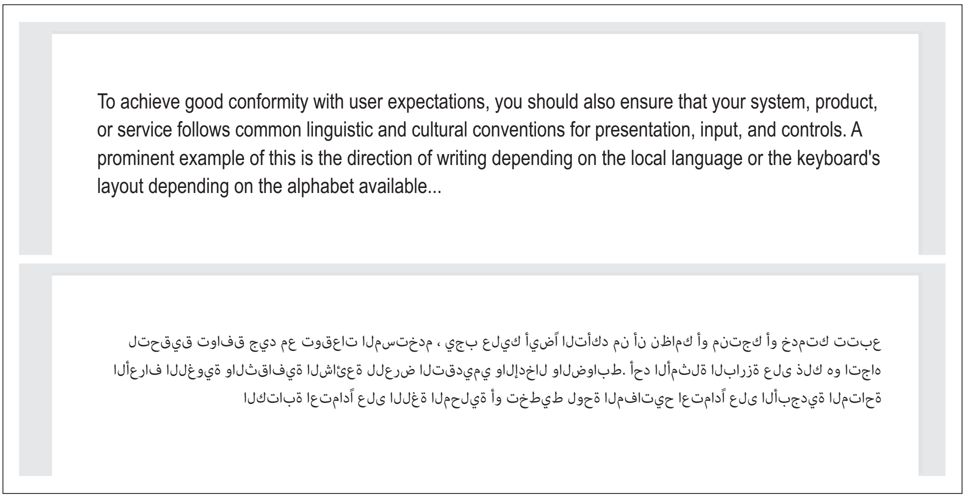
For example, if the software on a PC uses the same disk symbol and operating logic within the software to save results, this software is internally consistent. If it uses the same disk symbol and operating logic as the computer's operating system, it is also consistent externally.

The easiest way to ensure internal and external consistency is to make the process, controls, and wording comparable within and between applications. This goal also means that displays and controls that are used for different things should be designed differently. However, if you follow this rule all the time without exception, you may fail to implement obvious improvements (for example, the floppy disk icon remains, even though floppy disks hardly exist anymore). Thus, in this context, you need to carefully weigh and decide when conformity with user expectations is important and when the potential for improvement is so great that this conformity can be broken. Incidentally, this advice also applies when you deliberately design a system, product, or service not intended to adhere to common conventions. Examples include games or explorative applications that are designed to surprise users.

To achieve close conformity with user expectations, you should also ensure that your system, product, or service follows common linguistic and cultural conventions for presentation, inputs, and controls. A clear example of this need for conformity is how font direction (writing left to right, or vice versa) depends on the local language, as shown in Figure 7.83, or the keyboard's layout depending on the alphabets available.

The last aspect we listed is whether any changes in the context of use would need to be addressed.

These context of use changes include localization, that is, the ability for the system language to be adapted so that users speaking other native languages can also use your system, product, or service. Also, connected devices should be recognized by the technical system and handled appropriately—whether by displaying the information in a different resolution or using other operating elements such as on-screen keyboards.



To achieve good conformity with user expectations, you should also ensure that your system, product, or service follows common linguistic and cultural conventions for presentation, input, and controls. A prominent example of this is the direction of writing depending on the local language or the keyboard's layout depending on the alphabet available...

لتحقيق توافق جيد مع توقعات المستخدم ، يجب عليك أيضاً التأكد من أن نظامك أو منتجك أو خدمتك تتبع
الأعراف اللغوية والثقافية الشائعة للعرض التقديمي والإدخال والضوابط. أحد الأمثلة البارزة على ذلك هو اتجاه
الكتابة اعتماداً على اللغة المحلية أو تخطيط لوحة المفاتيح اعتماداً على الأبجدية المتاحة

Figure 7.83 Example display with a font running from left to right and from right to left.

Producing a system with a high degree of conformity with user expectations is not always easy because user expectations are made up of many different previous experiences that cannot be directly observed or inquired about. Moreover, 100% conformity is not always desirable and can lead to a lack of fun and, thus, to a poor UX. Therefore, just as with all other interaction principles, you'll need to measure precisely what weighs heavier for your particular case.

Learnability

An often underestimated interaction principle is the principle of *learnability*, which can be defined in the following way:

> *"The interactive system supports discovery of its capabilities and how to use them, allows exploration of the interactive system, minimizes the need for learning, and provides support when learning is needed." (ISO 9241-110:2020)*

Thus, learnability means that the system, product, or service supports users in grasping its use as quickly as possible. If you want to optimize your system, product, or service in terms of learnability, then you should ask yourself the following three questions, which adhere to the standard:

- How well can users discover the information and controls they are looking for?
- How easily can users safely explore the discovered information and controls?
- How much does the solution support users in learning the consequences of their interactions?

To optimize the *discovery of information and controls*, your solution should try to help users discover the most critical information and controls as efficiently as possible. You can achieve this goal, for example, by designing dedicated entry points for the user's most important tasks so that users can more easily find the appropriate information, as shown in Figure 7.84.

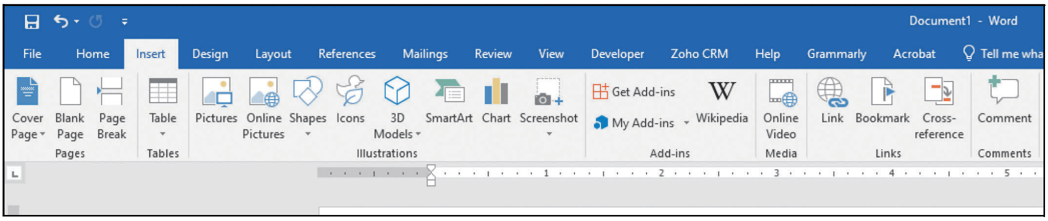
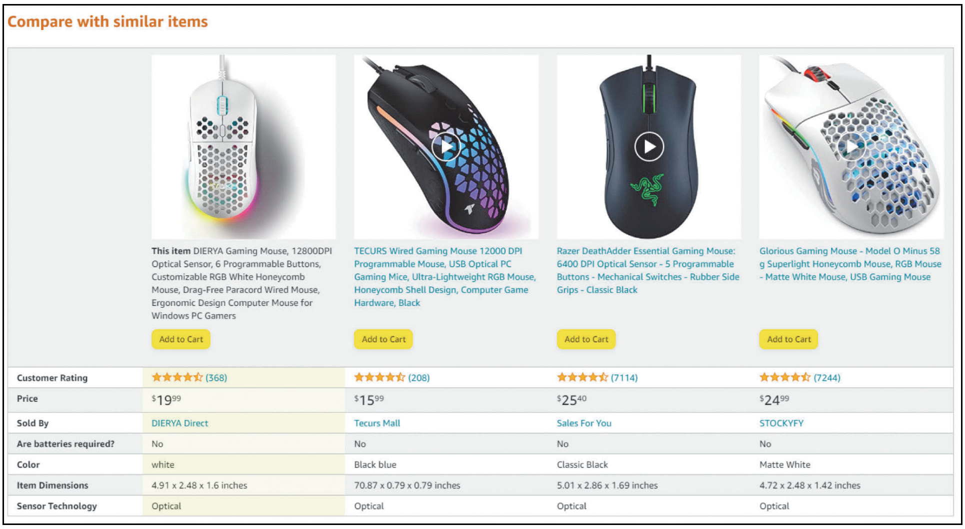

Document1 - Word
File
Home
Insert
Design
Layout
References
Mailings
Review
View
Developer
Zoho CRM
Help
Grammarly
Acrobat
Tell me wha
Cover Page
Blank Page
Page Break
Pages
Table
Tables
Pictures
Online Pictures
Shapes
Icons
3D Models
SmartArt
Chart
Screenshot
Illustrations
Get Add-ins
My Add-ins
Wikipedia
Add-ins
Online Video
Media
Link
Bookmark
Cross-reference
Links
Comment
Comments

Figure 7.84 Using jump-in points according to task to enhance discovery: Elements you can insert into a Microsoft Word document can be found under Insert.

To improve discovery, the way you deliver information plays a significant role. In this context, you'll be most successful if you structure information based on your user group's interests, tasks, and goals. Typical implementations of this concept include looking at things "other users are also interested in," where recommendations are made based on the behaviors of other users or the content currently being viewed, as shown in Figure 7.85.


Compare with similar items

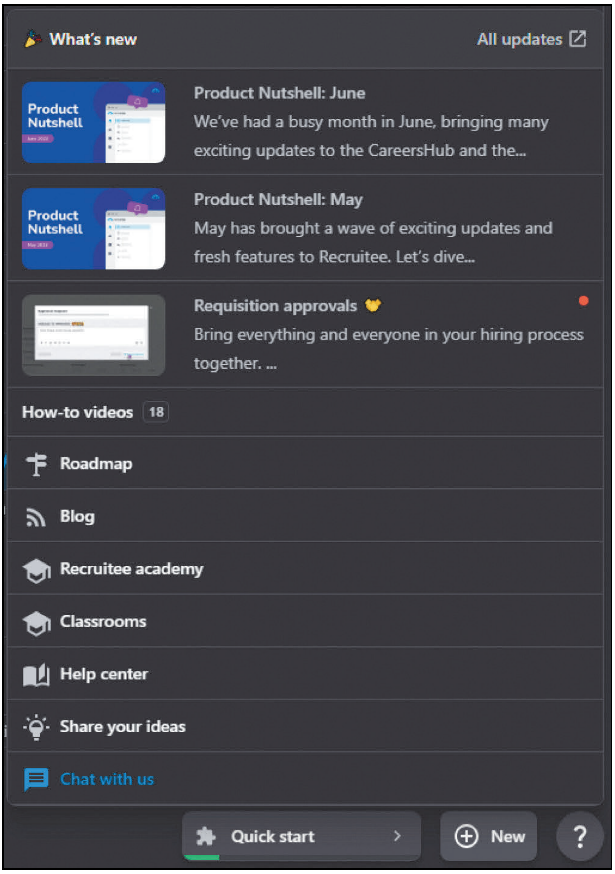
	This item DIERYA Gaming Mouse, 12800DPI Optical Sensor, 6 Programmable Buttons, Customizable RGB White Honeycomb Mouse, Drag-Free Paracord Wired Mouse, Ergonomic Design Computer Mouse for Windows PC Gamers	TECURS Wired Gaming Mouse 12000 DPI Programmable Mouse, USB Optical PC Gaming Mice, Ultra-Lightweight RGB Mouse, Honeycomb Shell Design, Computer Game Hardware, Black	Razer DeathAdder Essential Gaming Mouse: 6400 DPI Optical Sensor - 5 Programmable Buttons - Mechanical Switches - Rubber Side Grips - Classic Black	Glorious Gaming Mouse - Model O Minus 58 g Superlight Honeycomb Mouse, RGB Mouse - Matte White Mouse, USB Gaming Mouse
	Add to Cart	Add to Cart	Add to Cart	Add to Cart
Customer Rating	★★★★☆ (368)	★★★★☆ (208)	★★★★☆ (7114)	★★★★☆ (7244)
Price	$^{\$}19^{99}$	$^{\$}15^{99}$	$^{\$}25^{40}$	$^{\$}24^{99}$
Sold By	DIERYA Direct	Tecurs Mall	Sales For You	STOCKYFY
Are batteries required?	No	No	No	No
Color	white	Black blue	Classic Black	Matte White
Item Dimensions	4.91 x 2.48 x 1.6 inches	70.87 x 0.79 x 0.79 inches	5.01 x 2.86 x 1.69 inches	4.72 x 2.48 x 1.42 inches
Sensor Technology	Optical	Optical	Optical	Optical

Figure 7.85 Interaction patterns that increase exploration in a solution on Amazon.com.

If you use these and other interaction patterns to promote system discovery, you inevitably end up directly at the second point on the list you need to consider. Now, you need to think about whether the *exploration* is associated with any *negative consequences* for the users. As a rule, your system should allow users to explore without negative consequences. If avoiding this negative experience entirely is not possible due to the solution type, you can consider introducing alternatives that allow this. For example, you could include a sandbox mode where users can play with settings and functions or a simulation mode to explore the consequences of user actions without endangering live operations.

To make exploring the system, product, or service even more straightforward, having more than one access path for the important information and functions is recommended. For example, you can offer a dedicated menu item as well as a search field, thus allowing users to browse a solution freely.

If you want to further optimize your system, product, or service in terms of learnability, consider how you can show users the *consequences of their interactions*. One rule you can apply to achieve this goal is to provide feedback on a potential outcome before triggering a function. This feedback can be a simple confirmation box or feedback on the properties of the results. One common example is receiving feedback on a file's size before saving an image.


What's new
All updates
Product Nutshell
Product Nutshell

Product Nutshell: June
We've had a busy month in June, bringing many exciting updates to the CareersHub and the...

Product Nutshell: May
May has brought a wave of exciting updates and fresh features to Recruitee. Let's dive...

Requisition approvals
Bring everything and everyone in your hiring process together. ...

How-to videos 18
Roadmap
Blog
Recruitee academy
Classrooms
Help center
Share your ideas
Chat with us
Quick start
New
?

Figure 7.86 This system (from Recruitee.com) assists users in improving their interaction with the solution.

A learnable system supports users further to ensure that they learn about the solution and thus gradually gain a better understanding of performing their tasks. Further support is primarily about assisting users in such a way that they improve their skills in using the system. This learnability can be enhanced by providing tutorials or instructions and sample solutions. FAQs can also help users become better at using your system, product, or service, as shown in Figure 7.86.

Controllability

The next interaction principle we'll introduce is *controllability*, which can be defined in the following way:

> *"The interactive system allows the user to maintain control of the UI and the interactions, including the speed and sequence and individualization of the user-system interaction." (ISO 9241-110:2020)*

Controllability is, therefore, about giving users control over your system, product, or service. This aspect goes hand in hand with the fact that users can control the speed and the course of interaction with the solution and adapt it if necessary. If you want to check and improve the controllability of your solution, then according to the standard, you should ask the following questions:

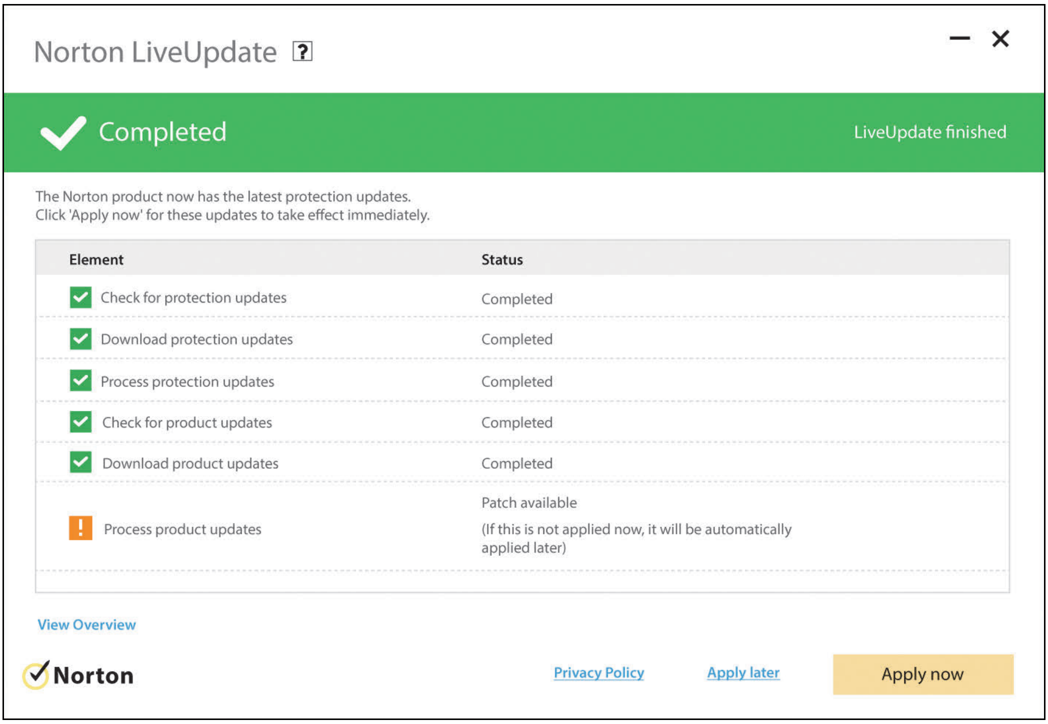
- How well does the solution support interruptions by users?
- How much flexibility does the solution allow users?
- How much individualization and customization does the solution allow users?

In general, you should try to design your solution in such a way that users can interrupt their currently executing tasks at any time. For example, users should be free to decide when to perform a system update rather than having updates start automatically, as shown in Figure 7.87.

You can make the controllability of your solution even better if you enable not only the interruption/suspension of tasks, but also the later resumption of the task. For example, if you make an online purchase but do not submit the order, good controllability would be provided if the products are still in the shopping cart a few hours later. Most e-mail programs also follow this principle: Unsent emails are saved as drafts and can be completed later.

You have achieved the ultimate controllability of interrupting tasks when users can interrupt and resume tasks and when you allow users to interrupt and manually resume automated tasks. An example is an automated vehicle allowing the driver to take control and continue manually.

You can further increase the controllability of your system, product, or service by increasing *flexibility* in interaction. This flexibility includes, for example, allowing users to perform the necessary steps in an order they prefer. In some areas, however, providing flexibility is very unusual.



The Norton product now has the latest protection updates.
Click 'Apply now' for these updates to take effect immediately.

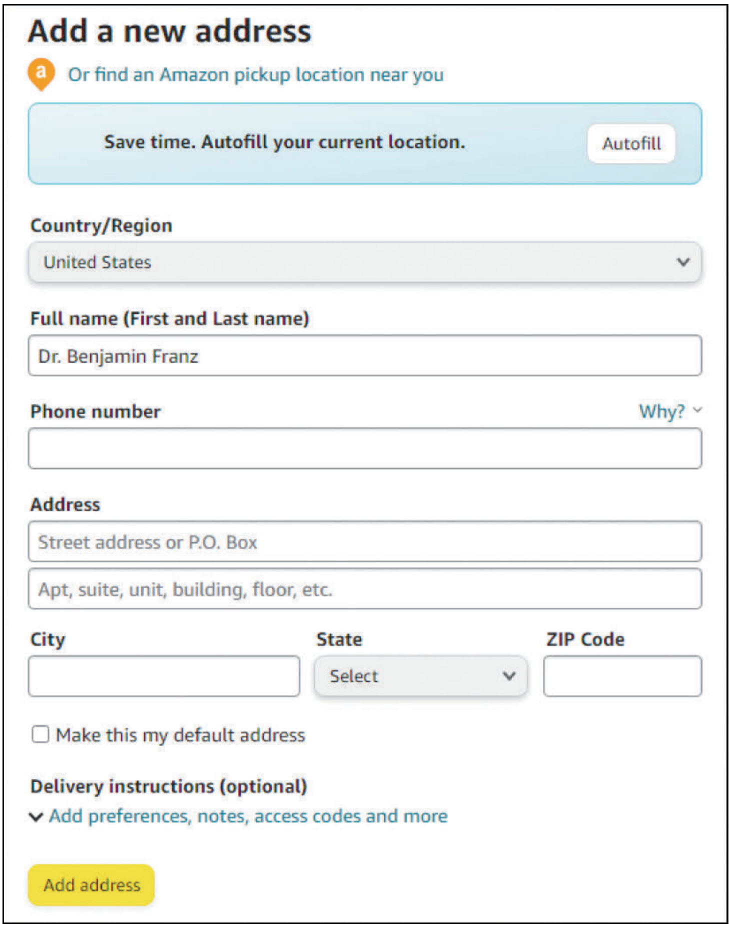
Element	Status
✓ Check for protection updates	Completed
✓ Download protection updates	Completed
✓ Process protection updates	Completed
✓ Check for product updates	Completed
✓ Download product updates	Completed
! Process product updates	Patch available (If this is not applied now, it will be automatically applied later)

Figure 7.87 Good controllability over updates from Norton: Users can decide whether the update should be applied immediately or later (automatically).

For example, when shopping online, you can't complete a payment before adding products to your cart. On a smaller scale, however, this flexibility is possible almost everywhere. Again, let's consider online shopping: When you type in a shipping address, usually a variety of required fields must be filled in, such as first name, last name, street, house number, zip code, and city. In this context, good controllability comes into play since the users are free to decide which field they want to fill in first, as shown in Figure 7.88.

Another way to increase the flexibility of your solution and, thus, its controllability is to give users a choice between different interaction methods. For example, you could enable your solution to be "switched" from keyboard and mouse input to voice or touch control.

But the interaction method does not alone contribute to a sense of—you guessed it—controllability, but also the ability to individually control the speed of interaction. By this speed, we mean that users can complete each step in the interaction when they are ready and are not forced, by external constraints, to take action quickly. To ensure this freedom, dialog boxes usually have controls that users can press to start the next step of the interaction. Common examples include **Continue**, **Finish**, **Next**, and **Back** buttons, as shown in Figure 7.89.



Add a new address

Or find an Amazon pickup location near you

Save time. Autofill your current location.

Country/Region: United States

Full name (First and Last name): Dr. Benjamin Franz

Phone number

Address

Street address or P.O. Box

Apt, suite, unit, building, floor, etc.

City

State: Select

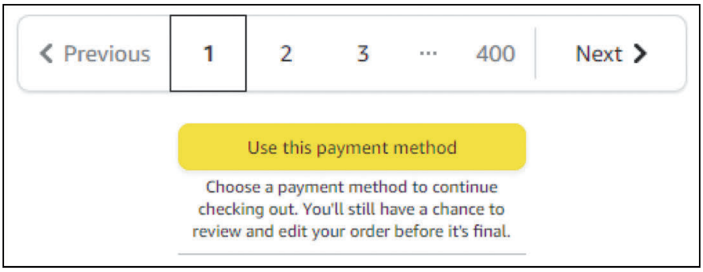
ZIP Code

☐ Make this my default address

Delivery instructions (optional)

Add preferences, notes, access codes and more

Figure 7.88 Increasing the controllability of a solution through flexibility in interaction on Amazon.com.

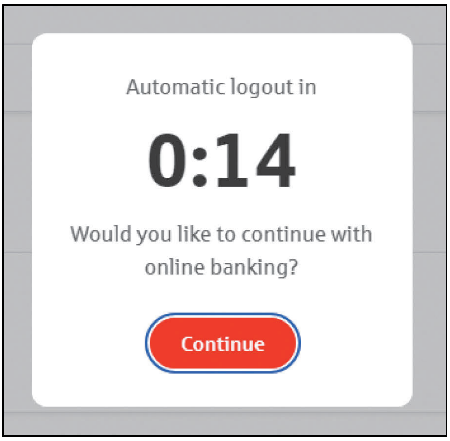


Choose a payment method to continue checking out. You'll still have a chance to review and edit your order before it's final.

Figure 7.89 Some typical ways to increase controllability in terms of interaction speed.

In some cases, however, the principle of controllability in interaction speed is also deliberately violated. However, you should have a good reason if you intend to do this.

Typical reasons for not allowing users to decide on the interaction speed are for security (e.g., automatic logout in online banking, as shown in Figure 7.90) or comparability (e.g., when completing interactive quizzes or exams).



Automatic logout in

0:14

Would you like to continue with online banking?

Figure 7.90 Interaction from Sparkasse in which parts of the controllability concerning the interaction speed were deliberately omitted.

A final point to ensure user controllability over interaction speed is to allow users to easily undo one or more previous actions, as shown in Figure 7.91. This undo function, for example, is now found in most office applications and is often so relevant that it usually has its own shortcut. By enabling undoing, you allow users to directly correct mistakes they have made and return to a previous state quickly and easily. However, one step back should be the minimum. We recommend allowing several steps backward, especially when the interactions become more complex and smaller.

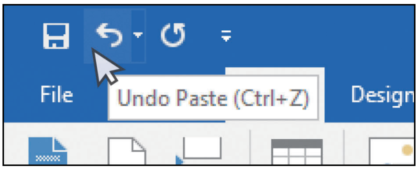


Figure 7.91 Improving controllability: the undo function in Microsoft Word.

The last area to increase controllability is to improve *customizability*, which can include, for example, allowing users to permanently adapt the default values prepopulated by the system to their own requirements. For instance, you could set the user's residence as the default location for a navigation system. Furthermore, you can allow users to customize the UI to suit their tasks and needs better. An excellent example is a configurable dashboard, where users can add widgets or click them away, as shown in Figure 7.92.

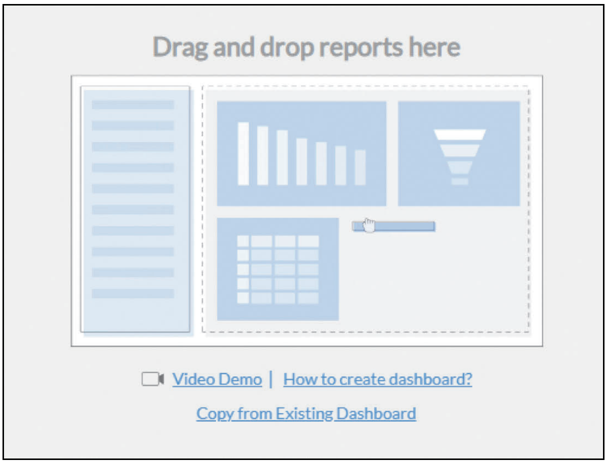

Drag and drop reports here
Video Demo | How to create dashboard?
Copy from Existing Dashboard

Figure 7.92 A customizable UI with widgets (Zoho One).

For the best controllability, however, you should not only allow the adjustment of the interface but, in the best case, also allow the resetting of the interface to default values and/or initial values. This option allows users to quickly and easily revise settings that were made incorrectly.

Use Error Robustness

Another interaction principle that we would like to present based on ISO 9241-110 is *use error robustness*, which can be defined in the following way:

> *"The interactive system assists the user in avoiding errors and in case of identifiable errors treats them tolerantly and assists the user when recovering from errors." (ISO 9241-110:2020)*

A system, product, or service can therefore be considered "error robust" if it supports users in avoiding errors or at least in limiting the damage caused by errors.

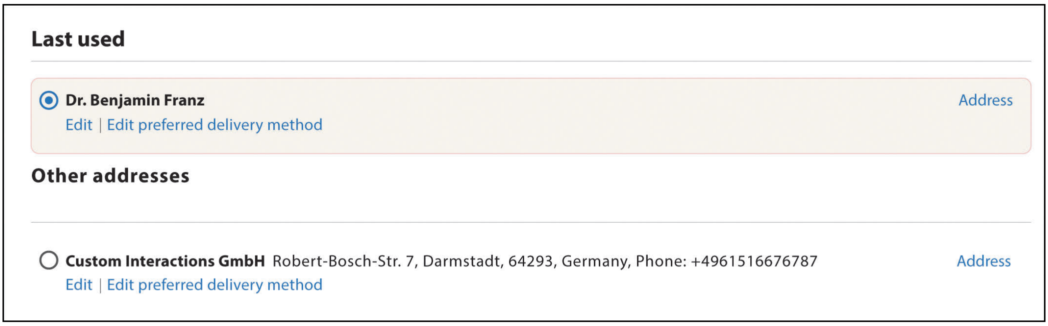
According to the standard, the following three aspects contribute to use error robustness:

- Does your solution help users avoid errors?
- Is your solution tolerant of errors made?
- Does your solution help users fix usage errors and/or mitigate their consequences?

To help users avoid mistakes, several rules should be followed according to the standard:

- One of the most important rules is that you can easily avoid mistakes if you do not force users to enter already-known information. For example, if you have an online store the users must enter their shipping address, you should assume that the billing address is the same unless a user says otherwise.

- The second rule recommends allowing users to select from existing data instead of entering new data, which is also aimed in a similar direction. This rule can also be explained well using online shopping as an example. Imagine you're registered with an online store and order there regularly. Needing to enter your address every time (instead of being able to select it from a list of saved shipping addresses) would probably lead to a poor UX on your side, as shown in Figure 7.93. In addition, the likelihood of errors increases compared to choosing between "successful" alternatives.
- Another way to reduce use errors is to allow only selections that are appropriate at the current time. A good example of this limitation is an appointment booking solution. In this case, only appointments in the future are offered and only for slots that are actually still free, as shown in Figure 7.94.



Last used

● **Dr. Benjamin Franz** Address
Edit | Edit preferred delivery method

Other addresses

○ **Custom Interactions GmbH** Robert-Bosch-Str. 7, Darmstadt, 64293, Germany, Phone: +4961516676787 Address
Edit | Edit preferred delivery method

Figure 7.93 Selecting addresses from an existing list on Amazon.com.

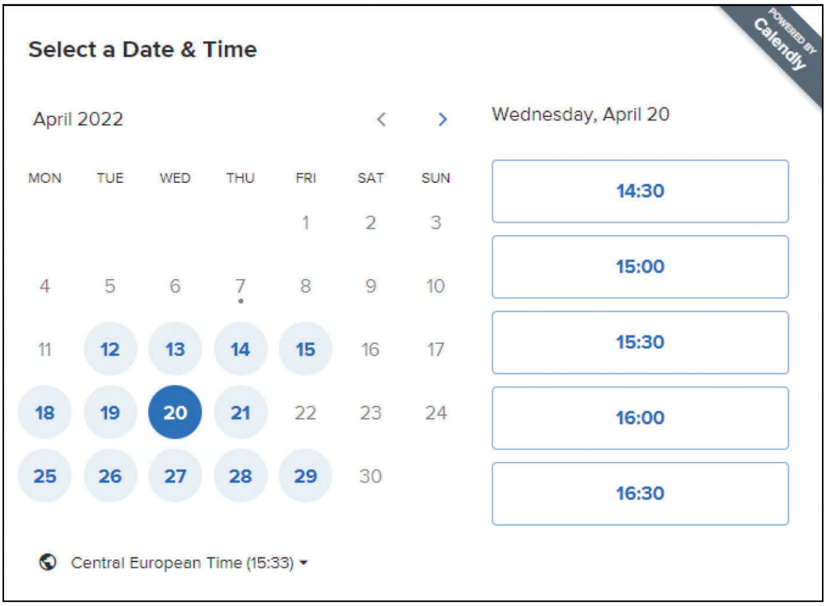

Select a Date & Time
POWERED BY Calendly
April 2022
Wednesday, April 20
MON TUE WED THU FRI SAT SUN
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
14:30
15:00
15:30
16:00
16:30
Central European Time (15:33)

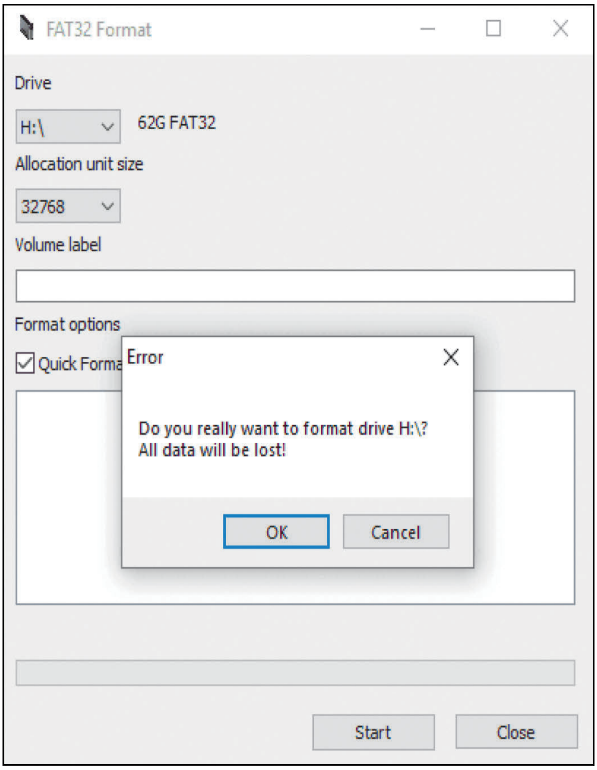
Figure 7.94 Increasing robustness against user errors by providing only working alternatives in a calendar tool (calendly.com).

A helpful approach is to also design your solutions in such a way that the work of the users is not lost due to an error (whether a technical error or an operating error). You can achieve this, for example, through automatic saving or versioning of content. At the highest level of error prevention, your solution detects possible user errors in advance or during use before the information is processed further.

To pull the example of online shopping out of our hat again, a good idea is to report back to your users when they enter an incorrect address that could not be found—ideally, with a suggestion for another suitable address. On the other hand, it would be a shame if you delivered the package only to realize that the address was incorrect upon delivery.

If a user action can lead to severe consequences, then you should also obtain confirmation from the user that this action is desired. A typical example is the confirmation dialog box before formatting a drive, as shown in Figure 7.95.

Once you have managed to prevent errors from occurring as much as possible, the next step is to increase your solution's tolerance for user errors. One rule to help achieve this tolerance is to design your solution so that users—even if they make a use error—can continue in the interaction without correcting it directly.



Drive: H:\ 62G FAT32

Allocation unit size: 32768

☑ Quick Forma

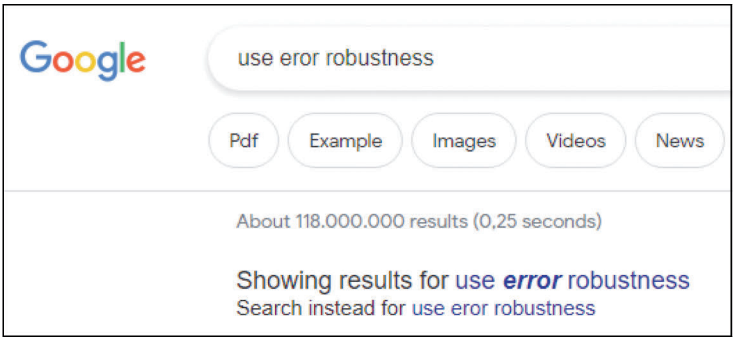
Error

Do you really want to format drive H:\?
All data will be lost!

Figure 7.95 Confirmation dialog box before formatting a drive.

Such an error could be, for example, a phone number that was entered in an incorrect format. Ideally, you should indicate the error directly to the user by outlining the incorrectly filled form field in red. However, you still allow the user to fill in the other fields before the phone number has to be corrected. However, at least when the form is submitted, you should actively point out errors to users.

Another way to increase robustness against user error is to use automatic correction. In this case, you should implement the correction to fix apparent errors for the users directly but still provide a straightforward way to override the error correction. A successful example of consistent implementation of this rule is Google Search. If you have a typo in your search term, Google will automatically search for the correct term in most cases but also offer you to "search instead for [your entered search term]," as shown in Figure 7.96.



About 118.000.000 results (0,25 seconds)

Showing results for use *error* robustness
Search instead for use eror robustness

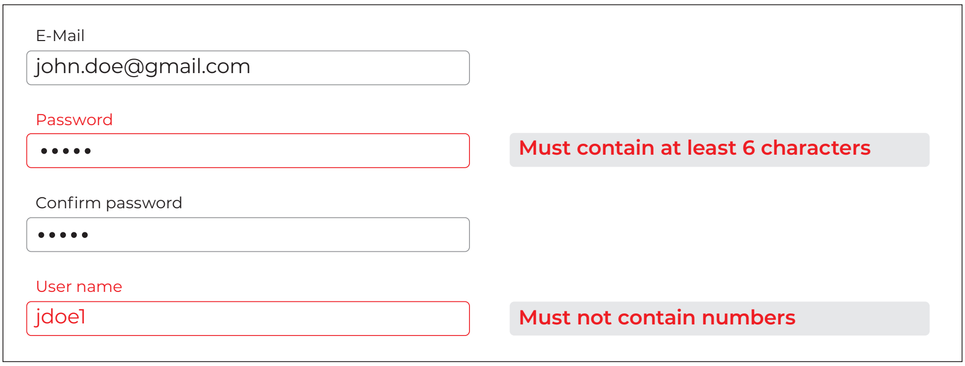
Figure 7.96 How Google increases use error robustness in its search engine by automatically correcting the search terms.

Sometimes, errors cannot be prevented, and even automatic correction cannot always be implemented successfully. Now, you should address how you can optimize your system, product, or service to support users in correcting mistakes that are made.

The first rule you should follow in this context is that you should always help users clearly identify, review, and then resolve any mistakes made in the first place. Specifically, you must clarify what is wrong, why it is wrong, and how it can be done better. Good design on a help form would highlight the field in question and explain to users what input is needed there, as shown in Figure 7.97.

Your error message should be formulated politely, precisely, and understandably. The more precisely you can describe the error, the easier it will be for users to correct it. So don't just say, "An error has occurred," but say, "The return date must be after the departure date." For a complex issue, a viable path may also be a brief explanation followed by the option to request more information via a link, box, or popup.

If you want to "wow" users now, make sure they know what needs to be done and make corrections as easy and quick as possible. For example, in an online form, you could jump directly to the incorrect form field when submitted so that users don't have to scroll and search for what's wrong. This advice is especially true when the feedback is out of the current field of view, and users must understand that they are supposed to search for an error.



E-Mail
john.doe@gmail.com

Password
•••••
Must contain at least 6 characters

Confirm password
•••••

User name
jdoe1
Must not contain numbers

Figure 7.97 Example error messages for an online form that not only indicates the error but also helps in correcting entries.

User Engagement

The last section on interaction principles is about *user engagement*, which can be defined in the following way:

> *"The interactive system presents functions and information in an inviting and motivating manner supporting continued interaction with the system." (ISO 9241-110:2020)*

Thus, user engagement means that the system, product, or service attracts the attention of users and motivates them to continue using the solution. According to the standard, to increase user engagement, you should look at three aspects in particular:

- Does the solution motivate users to use it?
- Is the solution perceived as trustworthy by the users?
- Does the solution increase user involvement?

Suppose you want to motivate users to continue using your system, product, or service. In that case, you should ensure that your solution informs users when tasks have been successfully completed and that no important, acute tasks are currently waiting. You'll be familiar with this type of feedback from various solutions. An example is antivirus software that shows your PC is protected, as shown in Figure 7.98.

Another way to increase the motivation of users to use the system, product, or service is the restrained use of *affective elements*. These elements in the UI include icons, images, and texts intended to trigger an emotional reaction in users. We're not always triggering positive emotions, such as joy; we can just as easily trigger (slightly) negative emotions such as fear and discomfort—depending on the goal of the design. In the simplest case, we could display images of smiling people near a support hotline number to set a positive mood for the conversation that will follow.

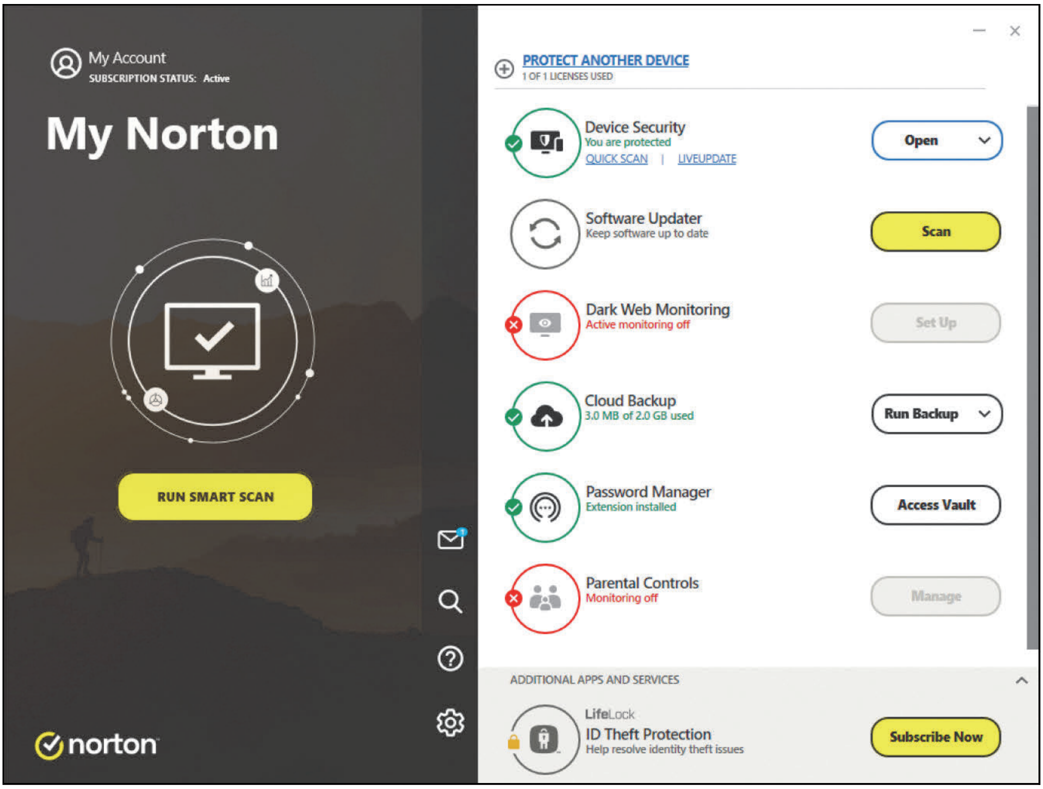

My Account
SUBSCRIPTION STATUS: Active
My Norton
RUN SMART SCAN
norton
PROTECT ANOTHER DEVICE
1 OF 1 LICENSES USED
Device Security
You are protected
QUICK SCAN | LIVEUPDATE
Open
Software Updater
Keep software up to date
Scan
Dark Web Monitoring
Active monitoring off
Set Up
Cloud Backup
3.0 MB of 2.0 GB used
Run Backup
Password Manager
Extension installed
Access Vault
Parental Controls
Monitoring off
Manage
ADDITIONAL APPS AND SERVICES
LifeLock
ID Theft Protection
Help resolve identity theft issues
Subscribe Now

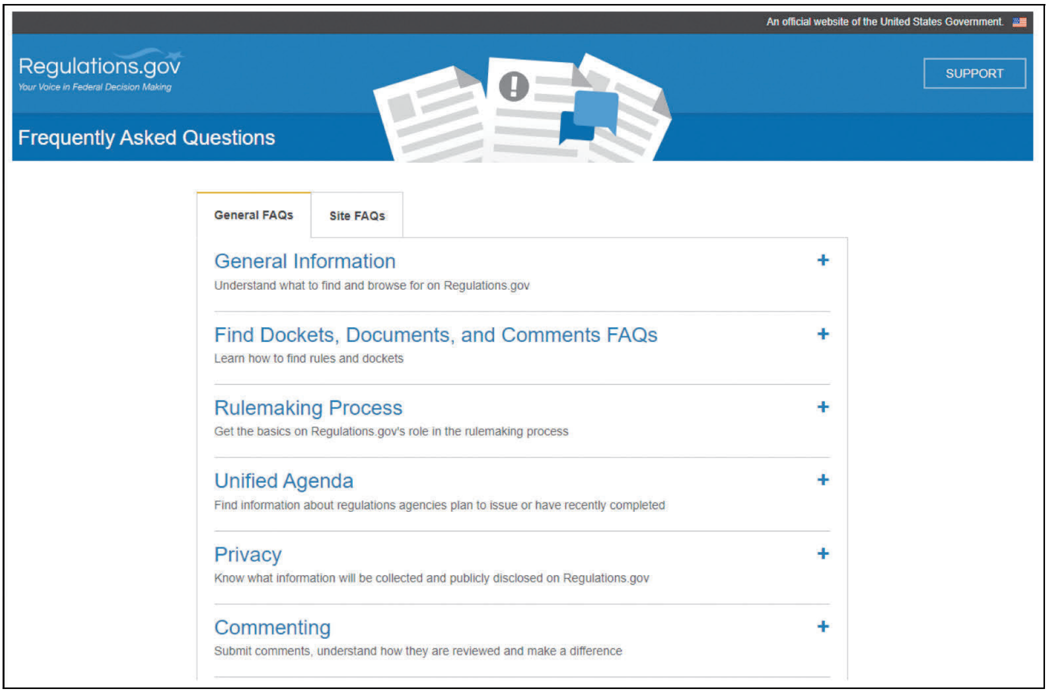
Figure 7.98 A Norton anti-virus program response designed to increase user engagement.

To maintain lasting motivation to use your system, product, or service, treating your users with respect is vital. This respect can be demonstrated in how you address users. To express respect, a helpful tactic is to use polite forms such as “please” and “thank you” in your communications.

If you want to work even further on increasing your users’ motivation, then you should try to anticipate and address needs that the users themselves did not even know they needed.

For example, suppose you have a ticket system and therefore know which requests users regularly bring to the support team. In this case, you can use this knowledge to increase users’ motivations to use your solution. For example, you could create an FAQ page that answers the most common questions, as shown in Figure 7.99.

As the saying goes: You never get a second chance to make a first impression. This motto is as true for systems, products, and services as it is for people. Accordingly, you should ensure that your solution makes an excellent first impression if you want to increase the motivation to use it.


An official website of the United States Government.
Regulations.gov
Your Voice in Federal Decision Making
SUPPORT
General FAQs
Site FAQs

Frequently Asked Questions

General Information +
Understand what to find and browse for on Regulations.gov

Find Dockets, Documents, and Comments FAQs +
Learn how to find rules and dockets

Rulemaking Process +
Get the basics on Regulations.gov's role in the rulemaking process

Unified Agenda +
Find information about regulations agencies plan to issue or have recently completed

Privacy +
Know what information will be collected and publicly disclosed on Regulations.gov

Commenting +
Submit comments, understand how they are reviewed and make a difference

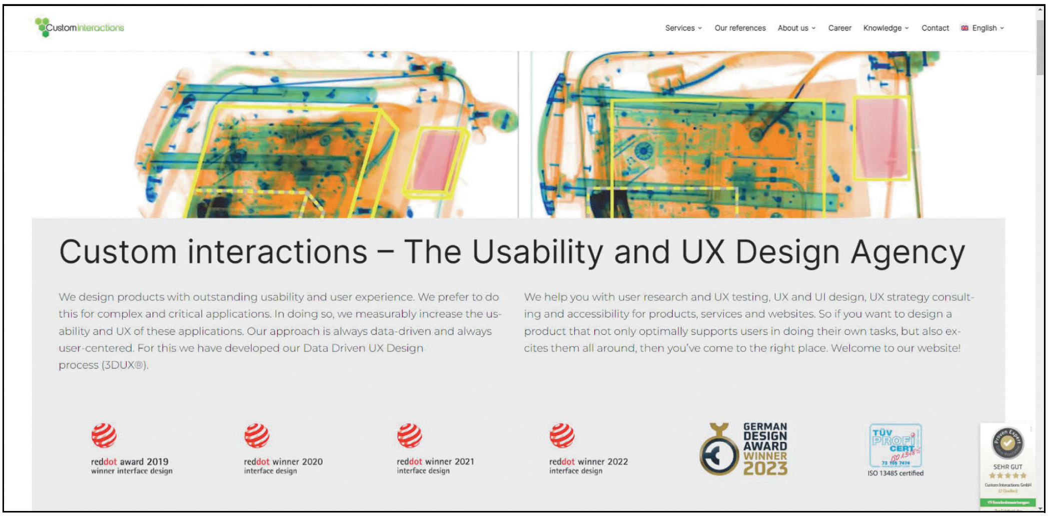
Figure 7.99 FAQ page on the Regulations.gov website.

What contributes to an excellent first impression depends on your solution and can be multifaceted. Overall, however, users should be able to find their way around at first glance, understand what they can do and how they can do it, and feel that the whole thing is presented in an appealing way.

Thus, you should ensure that your system, product, or service is designed to appeal to the intended target audience—and that this design does not reduce the effectiveness or efficiency of the solution. One way to achieve this goal is to have a mature visual design that matches the desired associations, both within the solution itself and in any accompanying materials such as packaging and instructions.

The last item on the do-to list to increase motivation to use your solution is to refrain from unrealistic requirements. One demotivating factor for many people is when they realize they cannot answer specific questions or complete a specific task. Imagine, for example, if you were asked not only for your birthdate but also the time of your birth to book a flight. Or you are asked not what year you left school, but on which day you took your last exam. Therefore, limit yourself to requirements that your users can actually (easily) fulfill.

Now, let's examine how you can increase user engagement by increasing the system's *trustworthiness*. The first step towards enhancing trust is to design the solution in such a way that trust is created. Examples include clearly communicating what benefits and risks are associated with using the system, product, or service. *Transparency* is the key word of the day in this context. From warning notices to sharing the company's history, including pictures of the founders, many things are possible to increase trust. Some other ways to increase trust include seals of approval, test results, work samples, and customer reviews, as shown in Figure 7.100.


Custom interactions
Services
Our references
About us
Career
Knowledge
Contact
English

Custom interactions – The Usability and UX Design Agency

We design products with outstanding usability and user experience. We prefer to do this for complex and critical applications. In doing so, we measurably increase the usability and UX of these applications. Our approach is always data-driven and always user-centered. For this we have developed our Data Driven UX Design process (3DUX®).

We help you with user research and UX testing, UX and UI design, UX strategy consulting and accessibility for products, services and websites. So if you want to design a product that not only optimally supports users in doing their own tasks, but also excites them all around, then you've come to the right place. Welcome to our website!

reddot award 2019
winner interface design
reddot winner 2020
interface design
reddot winner 2021
interface design
reddot winner 2022
interface design
GERMAN DESIGN AWARD WINNER 2023
TÜV PROFI CERT
ISO 13485 certified
SEHR GUT
★★★★★

Figure 7.100 Integrating seals and prizes to increase trust in our website.

Additionally, you should give users a realistic sense that using the solution is possible without *harm*. To achieve this aspect of trust, you have several options: For example, you could point out security protocols in place, require encryption, or make your revocation policy prominently public.

The last point to increase user engagement is to increase *user participation*. A common way to increase participation is to offer direct exchange and support between users, for example, through forums. Additionally, you should reward your users for their activities. The topic of level advancement or earning awards is particularly well known in gaming. However, these mechanisms are now increasingly being used in other contexts, for example, to highlight particularly active members in forums.

To ensure that your users feel well informed and involved, you can also inform them about product updates and explain upcoming changes in advance. It doesn't matter whether the news is about a new look or a change in functionality. In the best case, your users can decide whether and when they want to switch to the new version, as shown in Figure 7.101.

One side of user participation is informing users about upcoming changes in advance. However, you can further strengthen user loyalty by enabling users to submit suggestions for improvements or desired features. You can enable this feedback by implementing the rating of individual features on a scale; integrating a feedback form, for example, as shown in Figure 7.102; or by creating a hotline, chats, or user tests.

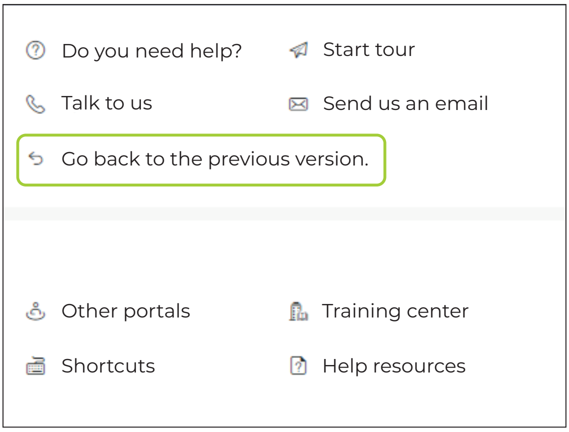

Do you need help?
Start tour
Talk to us
Send us an email
Go back to the previous version.
Other portals
Training center
Shortcuts
Help resources

Figure 7.101 This setting from Zoho One allows users to switch back to the old version of a UI.

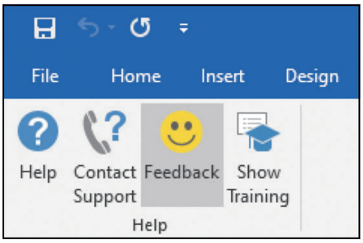

File
Home
Insert
Design
Help
Contact Support
Feedback
Show Training
Help

Figure 7.102 Integrated feedback form in Microsoft Word.

If you ask users to provide feedback and suggestions for improvement, you should also try to ensure high transparency, by which we mean that you give users feedback on development decisions, perhaps on an individual level by responding to individual submissions. But a common approach is publishing a list of the most requested changes or discussing upcoming releases in a developer diary.

The Interaction Principles in a Nutshell

In summary, let's briefly recap our discussion of interaction principles:

- **Suitability for the user's tasks**
  An interactive system is suitable for the user's tasks when it supports the users in completing these tasks.

  Key questions: Is the solution suitable for solving a specific task? How much effort is required to solve this task(s)? Are there defaults that help users successfully perform the task?
- **Self-descriptiveness**
  The interactive system presents appropriate information, where needed by the user, to make its capabilities and use immediately obvious to the user without unnecessary user-system interactions.

Key questions: Is all the necessary information present and obviously available? Is it clear to users what stage the process is at or what state the solution is in?

- **Conformity with user expectations**
  The interactive system's behavior is predictable based on the context of use and commonly accepted conventions in this context.

  Key questions: Does the system behave appropriately, and does it also provide appropriate feedback to the users? Is the system consistent internally but also externally? Have any changes arisen in the context of use that would need to be considered?
- **Learnability**
  The interactive system supports discovery of its capabilities and how to use them, allows exploration of the interactive system, minimizes the need for learning, and provides support when learning is needed.

  Key questions: How well can users discover the information and controls they are looking for? How easily can users safely explore the discovered information and controls? How much does the solution support users in learning the consequences of their interactions?
- **Controllability**
  The interactive system allows the user to maintain control of the UI and the interactions, including the speed and sequence, and individualization of the user-system interaction.

  Key questions: How well does the solution support interruptions by users? How much flexibility does the solution allow users? How much individualization and customization does the solution allow users?
- **Use error robustness**
  The interactive system assists the user in avoiding errors and, when identifiable errors arise, treats them with tolerance and assists the user when recovering from errors.

  Key questions: Does your solution help users avoid errors? Is your solution tolerant of errors? Does your solution help users fix usage errors and/or mitigate their consequences?
- **User engagement**
  The interactive system presents functions and information in an inviting and motivating manner supporting continued interaction with the system.

  Key questions: Does the solution motivate users to use it? Is the solution perceived as trustworthy by the users? Does the solution increase user involvement?

Having described the basics in this chapter, we'll now turn to the data-driven UX design process in the following chapters, starting with the preparation.

Chapter 8

Data-Driven UX Design in Detail: Preparation

“Tell me how a project starts, and I’ll tell you how it ends.”
—Gero Lomnitz (consultant, trainer, and coach)

In this chapter, we’ll detail the individual steps of the data-driven user experience (UX) design process. We present the process and procedures as we have compiled and developed them for our own work. In our work experience, we have become acquainted with various approaches from design literature and science over the past 20 years, tested them in practice, and adapted these methods for our purposes.

All of the coming steps have been proven to work for our field of activity—usability and UX design in complex, medical, or critical environments—and for us. Feel free to adapt the parts that fit you and use them as a basis for your own approach.

When we say the word *planning*, two factions quickly form:

- On the one hand, some strategists love planning and want to have everything prepared down to the last detail. Often, in addition to plan A, these individuals have a plan B and a plan C, maybe even up to plan Z, ready to go.
- On the other hand, for the creative spontaneous types, planning is often an imposition. These individuals often assume that planning changes anyway, and therefore, it doesn’t make sense to plan things.

We’re both on that spectrum, too, and one of us at each end—at least until a few years ago. In many joint projects, we have moved towards each other, trying to combine the best of both worlds. The result could be called *semi-structured planning* (borrowing from *semi-structured interviews*, of which you’ll hear more in this chapter).

By *semi-structured planning*, we mean you should keep the procedure’s objective in mind and divide the process into small steps in time. However, planning remains somewhat on a meta-level and only becomes concrete at certain necessary points. In this context, our approach is often “timely follow-up.” In other words, we roughly determine the schedule and define the objective. Then, we transfer the objective to the schedule but still leave out all the details that take a lot of time to determine and that may change repeatedly during the project.

As we progress in the project, we gradually flesh out the open points that need clarification. We define some critical points in the schedule as *points-of-no-return*. A typical example of a point-of-no-return is the start time for inviting study participants: As soon as we start inviting people, no longer can schedule changes be made for the survey or in the target group. Desired changes can only take effect in the next step. And so, we move through the project piece by piece, making the planning more concrete along the way. Just the content planning, mind you! In our projects, time planning is usually fixed from the outset and is only rarely postponed.

No matter where you are on the spectrum, because usability and UX design requires the interaction of multiple parties, you usually can't get by without planning. In this chapter, we'd like to use the concepts and mindset we've presented so far to show you how to plan your data-driven UX design approach meaningfully, even though so many unknowns are involved.

8.1 Planning the Data-Driven UX Design Process

The first step to solid planning is always to remember your target. To clearly define this target for yourself—but above all, with all your stakeholders too—we always advise starting every new project to follow a data-driven UX design approach with an internal stakeholder workshop.

8.1.1 Who Should Participate in the Stakeholder Workshop?

For us, running this workshop means that you bring all the people from your company to the table who have a legitimate interest in the outcome of your project and/or can positively influence it with their experience, knowledge, and relationships.

"Positive" in this context does not mean that you only bring people to the table who are excited about the idea. "Positive" means that the person adds value to the planning of the process or the implementation of the product, including harsh critics: Especially when developing internal solutions, we even recommend involving likely critics from the beginning.

In most companies, you'll now think of quite a few important people, especially if you have many levels in your company's hierarchy. Our recommendation is to try to get people at different levels of the hierarchy excited about participating in the workshop. Everyone who can make a valuable contribution should be included. In our experience, perspectives differ significantly between management and other positions. If you can, you should try to include all the different views. The more decision-makers supporting the project, the easier it becomes. The following rules have proven helpful to us in selecting participants:

- The person making the budget decision should be involved.
- (When developing internal solutions): At least one person who will have to work with the solution later on should be involved.
- (When developing external solutions): At least one person (preferably two) with regular direct customer contact should be included—but we do not include external customers in the stakeholder workshop.
- The core team responsible for development should be present, with one person for each aspect.
- At least one person from each of the following areas should be represented:
  - Distribution/sales
  - Marketing
  - Product management/strategy
  - Legal/quality management/logistics (depending on the product)

Conducting a Stakeholder Workshop in Hierarchical Companies

Especially in companies where the data-driven UX design approach is still new and strong hierarchies prevail, the approach to planning can make the difference between success and failure. We've had a lot of success getting all the areas involved on board as early as possible. If you lack contact with some areas, we recommend that you start presenting your project at the highest possible level that still has a connection to the topic.

Work your way down if you find no willingness to participate in a workshop at that level. This step also allows you to ask who the best contact person is. You'll then find out relatively quickly whether a willingness to participate exists and at which level. In your hierarchy, you should, of course, start with your supervisor.

A crucial goal is to have everyone at the table at the workshop who will also see the project results later and can easily revise them based on gut feelings or lack of participation. We sometimes jokingly call this tactic playing to the *highest paid person's opinion (HIPPO)* in the company. Even though we'd like to tell ourselves otherwise: In most cases, some decision-makers will have the power to question project results just like that. These people need to be involved initially and kept informed as the project progresses.

Of course, not all projects have HIPPOs, and the strategic decision may rest entirely with the team. However, a helpful activity would be to use the workshop and its preparation to clarify expectations.

8.1.2 The Agenda of the Stakeholder Workshop

In the stakeholder workshop, we recommend several activities:

1. Clarifying the order
2. Collecting existing knowledge
3. Resharpening the job
4. Discussing the organizational framework
5. Clarifying the next steps

This structure will probably seem familiar if you already have experience managing projects. The first thing you need to do at the beginning of the project is define the project's framework.

Clarifying the Order and the "How-Might-We" Question

This step is about ensuring that everyone in the room has an equal understanding of what they want to have achieved at the end of the project. We have had good experience with not just saying what we intend to do but asking each person in the room to define their project expectations. For us, this discussion also means asking them to tell us what would make a truly outstanding project for the person in question. This approach is the only way we can align our activities appropriately at the very beginning or manage expectations at an early stage.

We've often discovered expectations that we simply can't meet at this stage. After all, they are not the focus of the project or not the responsibility of the usability and UX design team. In our initial stakeholder workshops, what surprised us the most was that eight people in a room, who all knew about the project, all had different expectations. By now, we are used to this diversity because it happens regularly, which is why holding stakeholder workshops is even more critical.

When you have many divergent opinions, deciding for each point whether they should be part of the project is imperative. We usually do this assessment again for all to see, repeat the decision, and have everyone agree again that the decision aligns with the common understanding.

One way to capture the essence of a project is to pose a *how-might-we question*, which we introduced to you in Chapter 4, Section 4.2.3.

"How-Might-We" Questions

The idea behind *how-might-we* questions comes from design thinking and pursues the goal of formulating the mission in a single question. Using this one question makes it much easier during the project to keep the focus on the core topic and the question: "Does this still fit with what we wanted to do?"

In addition, this rather compact formulation also always allows you to compare whether the goal still seems reasonable based on the data. In this way, the question helps you define a termination condition—something that is important in many cases but often not thought about.

The *how might we* question gets another bonus point from us because of its wording: Its solution-oriented view switches your gaze from the problem view ("Everything is bad, and nothing works") to the solution view ("How can we make sure that...?"). This switch lets your brain become active in a completely different way than solely looking at the problem.

Therefore, the formulation of your project assignment should follow the following structure (*how might we* question):

> *How can we help* ______________________ *[who?] to* _________________________
> *[do what] if/when* ______________________ *[in what situation?]?*

Thus, you can formulate a question where you define both the target group and the support you want to provide with your solution and the context in which this support is provided. This step also makes it clear to everyone in the room that you need a target group and a problem to develop a successful system, product, or service.

Once you have formulated this *how-might-we* question, write it down and take it with you for the rest of the project. Perhaps later, the data will lead to the need to adjust or resharpen the question again, but in any case, a clear goal now exists for everyone.

The next step is to collect existing knowledge.

Collecting Existing Knowledge

The second step is to gather, from various sources, the organization's shared knowledge of the project goal. This research also includes collecting what is not yet known. In the best case, you already have collected, sifted, and prepared in writing the existing knowledge from the organization before the workshop. If not, then our recommendation is to focus on the knowledge available in the room and, if necessary, to record knowledge available from other people subsequently.

In this part of the workshop, depending on how well prepared you are, you can proceed according to the recommendations described in Chapter 6, Section 6.1.2, and draw up an assumptions-knowledge matrix, as shown in Figure 8.1.

At the end of this step, you'll have a good overview of what you know as well as the open flanks that can bring you and your project down. We now have the perfect starting point to launch into the first step of the data-driven UX design process. In this step, after all, your task is precisely to close knowledge gaps.

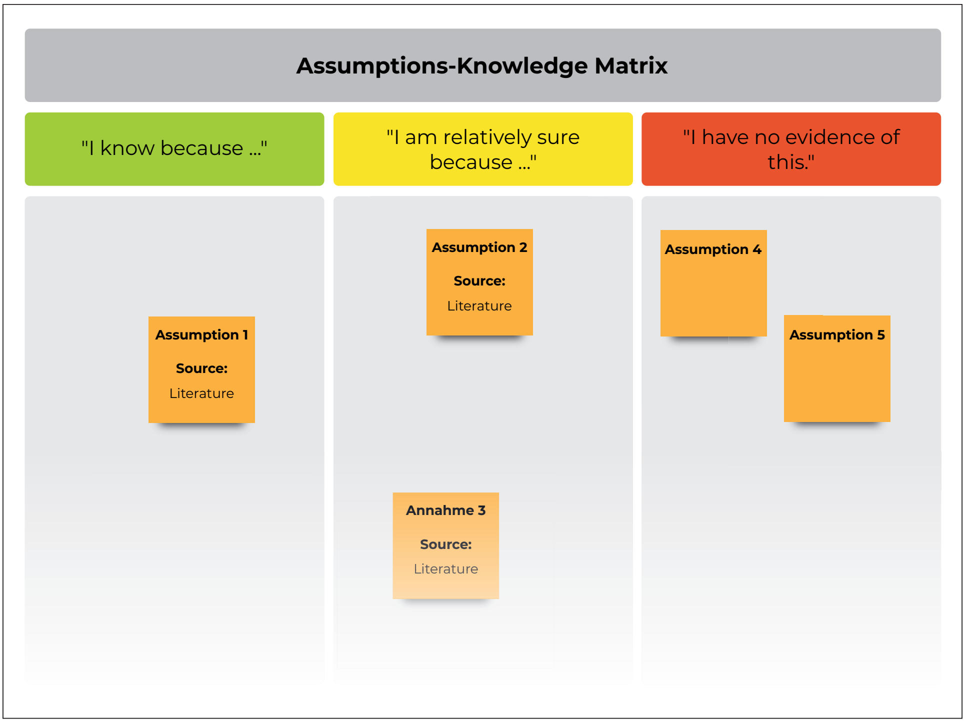

Assumptions-Knowledge Matrix
"I know because ..."
"I am relatively sure because ..."
"I have no evidence of this."
Assumption 1
Source:
Literature
Assumption 2
Source:
Literature
Annahme 3
Source:
Literature
Assumption 4
Assumption 5

Figure 8.1 Assumptions-knowledge matrix as presented in Chapter 6.

Now that the cards are on the table and it's pretty clear what's a fact, an assumption, and what's an unknown, you should go back to mission clarification and put your *how-might-we* question to the test.

Resharpening the Job

This part can be long or short, depending on the current status of the project. In the best case, all the knowledge in your organization has led to the feeling that you (and your stakeholders) are on the right track with the project. Then, a simple question is usually enough: "Now that we see all this in front of us: Do you think our *how-might-we* question is still right?" If nothing surprising has happened, only a brief discussion will ensue at this point, or you may just see heads nodding.

If not, you should start over again and will need to decide the project's new direction based on the data. And then—as you've probably figured out—you can formulate a new *how might we* question, which then stays with you until you've completed the project (or you've learned that this question is not valid either).

Discussing the Organizational Framework

Then, when both the mission and knowledge gaps are apparent, you've already arrived at the organizational framework. Again, the keyword in this step is *expectation management*, especially with regard to the following goals:

- Make it clear that the work will be iterative and human centered.
- Lay out all the steps for all stakeholders and clearly state that you plan to stringently align with user expectations to ensure a successful project.

We recommend that the focus in this discussion is also once again clear that we are now gathering *evidence* that the project is on the right track. Most of the time, we also explain again in detail what such a cycle might look like and make it clear that, from our point of view, evidence is only evidence if it comes from real users or buyers and not from colleagues, superiors, or similar people.

Then, it's really about *organizational fine-tuning*. The more highly staffed your stakeholder workshop is, the shorter this part "in the round" should be. Most of the time, we only decide on individuals who are directly involved in consultations, who are part of the project advisory board, who are on a distribution list, or who want to withdraw entirely from the project as it progresses.

In this way, we can ensure that everyone is sufficiently informed and that the opinions and feedback of the decision-makers are incorporated into the project. This approach also enables us to maintain the speed of the project because scheduling and approvals become easier.

After this discussion, we can move on to the last part of the stakeholder workshop: the *next steps*.

Clarifying the Next Steps

Since you'll have learned some things in the workshop that were not on the table before, we recommend that you perform the following steps next:

1. Document the key outcomes (e.g., by taking minutes that record the key decisions of the workshop) and distribute this documentation to the people involved.
2. Develop a project plan based on the newly acquired knowledge. The level of detail of the project plan depends on the way of working and can range from sprint plans to fully defined project plans.
3. Operationally start the project with the first work package (e.g., with the execution of the first sprint in agile design).

One goal applies to all these steps: Make it clear who is taking on a task, when the task is to be completed, and when you need feedback from the group. We also recommend that you agree with all participants during the workshop that a lack of input by the feedback deadline will be interpreted as tacit agreement to the arrangements. This understanding will help you maintain the project's pace and ensure commitment.

Transparent Communication

One more thing we would like to emphasize: Your ultimate goal should be transparent communication. No matter what happens in the project, avoid surprising anyone. In our experience, it's not so much that a project runs perfectly smooth from the first second but that everyone involved feels that the project is moving in the right direction and being followed up.

This feeling is usually created proactively and transparently by informing all stakeholders about the current status of a project, any emerging issues, and next steps and by not "accidentally" stumbling upon problems.

The documentation of the stakeholder workshop is the basis for the project and should therefore be particularly clearly formulated. To achieve this goal, we recommend writing the documentation calmly after the workshop.

But feel free to communicate what people can expect in the documentation: We document the complete workshop process with the core results for each step. We also explicitly note the exclusion of certain activities from the project—including the reasoning found as consensus in the workshop.

If any open points still need to be clarified, these points will also appear in the protocol with responsibility and schedule assigned. You can then plan the project in detail. You now know some of the unknowns and open flanks you need to tackle, and you can estimate how long the individual steps will take. Usually, the estimate will be pretty good, and depending on whether you need to plan your process against a *quality goal* or a *time goal*, you can either define a fixed *end time* or define *key performance indicators (KPIs)* that you'll use to determine that you're done. (For more on defining end times, see Chapter 13; for more on metrics, see Chapter 3.)

Finally, in terms of "next steps," you should determine when you would like to start the project operationally. We recommend that you clearly communicate this start time so everyone involved is committed to a schedule. If you only say something like "We'll start as soon as we have received all the feedback," our experience shows that you'll need significantly longer than necessary.

Concluding the Stakeholder Workshop

At this point, you can already close the workshop in the round. We usually spend a little more time with the direct project team afterward. Then, it's a matter of clarifying the other organizational framework conditions by asking, for example, the following questions:

- Who releases results (whether intermediate or final)?
- How much time do we need to allow?
- Who is not available when?

These organizational points must be clarified—but only in the project team with which you work daily. Clarifying these issues in the grand rounds leads to resentment, which you can easily spare yourself.

Planning after the Workshop

We recommend that you always base the actual planning on time. In other words, plan your project in *sprints* with a specific duration and scope of work—based on the *scrum method*—but without the idea that an executable piece of software must be the result after each sprint.

Another way to plan the project based on timing is *rolling wave planning*. In this method, you plan a certain number of sprints or work cycles at the beginning of a project (for example, the first six sprints). Then, you start the first sprint, and as soon as you reach its end, you revise your planning based on the newly acquired data and plan one more sprint so that the next six sprints are planned again, only based on the new data and on new knowledge.

No matter what you call it: We recommend that you think about when you want to have reached a particular milestone and view sprints as pathways to those milestones. Then, at the end of the sprints, looking back at your achievements will always give you opportunities to consult with your stakeholders or the project advisory board.

Whether you go through the cycle once in its entirety during a sprint, only perform individual steps from the data-driven UX design process, or go through the entire process multiple times certainly depends on your current situation, the pace of feedback in the company, and the available resources.

In general, you can successfully combine complete short, data-driven UX design cycles and only partial cycles with sprints. However, especially if you're working in a dynamic environment, running through the iterations as quickly as possible regarding usability and UX design makes sense.

You might be wondering now why we distinguish between a sprint duration in agile processes and the lead time of an iteration in the data-driven UX design process. This distinction mainly concerns the embeddedness that we often experience in larger projects. We dedicate a separate section because the topic is so relevant, and we are repeatedly asked about it.

8.2 Integrating Data-Driven UX Design into Agile Development Projects

Although a crucial element, usability and UX design does not stand alone in space and will not work without other processes. In the early stages of human-centered (or as called in the past, *user-centered*) development, our discipline was always something like the iteration before the waterfall.

Waterfall Model versus Agile Development

In the *waterfall model*, systems, products, and services are thought of as linear. A list of requirements is formed at the very beginning of the development process. This list summarizes all the requirements for the solution, which are then implemented piece by piece.

This approach contrasts with *agile development approaches*, which start with a specific set of known requirements and then add to them piece by piece during development. The agile approach is, therefore, particularly compatible with human-centered and iterative design approaches, which aim to uncover unknown requirements as early as possible during development.

Agile approaches consider that all the requirements for the solution are rarely fully known at the beginning of a project, thus allowing for more flexibility and uncertainty.

Integrating usability and UX design was still relatively simple in a waterfall world. Before the technical development of the product began, an iterative human-centered process was set up to uncover all requirements. This process was assigned a set duration (for example, 3 months), and it was determined that all requirements would then be known. After the completion of the preliminary phase of 3 months, the “actual” development started. This approach rarely works today.

Over the last few years, the agile approach has found its way into many companies that offer digital products, systems, or services in the broadest sense. An agile approach considers that software projects are often challenging to plan and that discoveries are usually made during development. In agile processes, sprints are scheduled with a fixed duration, at the end of which the goal is a deliverable *increment*, that is, some part of the software that can immediately be adopted into live operations.

Thus, instead of a few big releases, more smaller releases characterize the development. This approach sounds tempting, and in many dimensions, it is. But we keep encountering one question: “How can we fully exploit the potential of usability and UX design if we think in sprints and therefore have no defined requirements phase in advance?”

We don’t have a standard response for you at this point. Still, we can tell you that, in practice, the answer often depends on the organizational framework conditions in individual companies.

In our experience, when we come in at the very beginning of the project and are involved in the planning ourselves, we often ask to run one or two sprints ahead. In this way, we can ensure the first implementation starts about 4 to 6 weeks later than we do. If possible, we use the first sprint to understand the problem, develop the first concepts, and collect the first user feedback. We use this knowledge in the *second sprint* to directly start revising the concepts according to the feedback and, in parallel, prepare the next topic that will become relevant afterward. Which topics are relevant is, of course, decided by the whole project team. With this approach, most topics are created, tested once, revised, and technically implemented in the next sprint. In this way, we can ensure that we can also permanently incorporate user feedback in ongoing development.

For this approach to work, however, some preconditions must be met, such as the following:

- You must prioritize features.
- The UX team must be able to iterate significantly more quickly than the time required by the sprints.

Prioritization must be established because, otherwise, the UX team will find it difficult to prepare the topics that should find their way into development. Of course, priorities can change from time to time. But if the rough planning is not even fixed, your UX team may work on features that only find their way into the product much later. While this early development wouldn't be a bad thing in itself, the problem is that the development team is now designing parts of the system themselves without input from you and your team. Although this situation can work, it doesn't always. Above all, this scenario should be avoided because the potential for collaboration is lost.

The second prerequisite concerns your *speed*. As you might have gathered from the description of our approach, we usually try to go through the data-driven UX design process once completely per customer sprint at a minimum. Thus, one development sprint would correspond to all four phases in the data-driven UX process. Or even eight phases if the process is run through twice. Suppose we couldn't do that and instead covered each step in one sprint (e.g., only doing the context of use analysis in the first sprint). In that case, we'd be back to the same situation as earlier: The development team would have no input in certain areas and simply decide "on their own" how to implement things. Again, this situation can work but doesn't always.

Ultimately, the challenge you face is to embed yourself in a dynamic and self-sufficient system. In our experience, we bring the most significant possible benefit when our work provides a head start for the development team. We either get that extra time from the project (the one or two sprint lead) or earn it by working faster. In the best case, however, both happen. The focus will always change as other functions previously on the back burner move into the foreground. You often handle these changes by conducting *user research* and uncovering insights that shift the focus.

However, some sprints in between will focus exclusively on things “under the hood,” and you and your team can only make a small contribution. In these sprints, you can gain additional time you’ll probably lose elsewhere—all of which is normal.

If you’re involved in planning, the less flexible your product and the more critical the application, the more of a head start you should try to get before implementation. Consider again our flexibility and risk matrix from Chapter 5, Section 5.3, as shown in Figure 8.2.

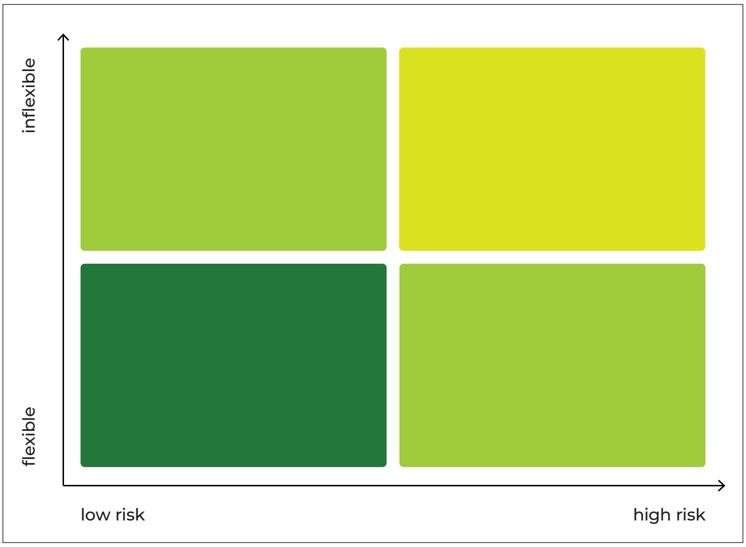


Figure 8.2 Flexibility and risk matrix for determining the required data in the data-driven UX design process.

The further up to the right you’re with your system, product, or service, the more confident you must be that you’re correct with your design. Of course, applying agile development methods to inflexible products is only possible to a limited extent anyway. But not impossible—for example, Tesla has challenged the car industry with its over-the-air updates. So, if you make an inflexible product more flexible by using a “trick” (e.g., adding or unlocking product features later through software updates), applying agile methods can also make sense here because you would need less of a head start.

Even if you’re sure of an extremely high degree of flexibility in your particular case, and can therefore start the implementation in a quite relaxed manner, we recommend reflecting once again at this point on the application you’re developing. The focus should be on the level of risk. The greater the potential risk is (considering the areas of *risk to people, economic risk to the company*, and *risk to the company’s reputation*), the more closely you should look at the interface beforehand and the more confident you need to be (see Chapter 5, Section 5.3.2). So, a higher risk could also lead to the decision to need more of a head start for usability and UX design before development.

If you don't have a high-risk product, a second option is available: *live testing*. The idea of agile development processes is to have an executable piece of software that can go live after each sprint. You may have the first development phase behind you and have a working product available on the market but only with a limited range of functions—what is called a *minimum viable product (MVP)*. Then, your goal might be to expand the range of functions as quickly as possible.

In this case, as a usability and UX designer, you can focus entirely on designing the new features and workflows during a sprint and ignore testing for now. Then, when the product has gone live (in the same or upcoming sprint), you can test quality live with real users. By taking this approach, you gain time for design and can thus implement new features more quickly. Of course, you could have been mistaken with some of your design choices. In fact, you might be completely wrong on some points. Therefore, you should only use this approach if the user group will not resent change.

Tip: Reduce the Risk of New Features in a Targeted Manner

Suppose you want to minimize the risk of being completely wrong. For this goal, we recommend that you always announce new and untested features as *beta features* (but replace this term with one your target audience understands). In this way, you can clearly communicate that these are still features in development. At the same time, you can arouse curiosity about new features and maybe even get more feedback earlier. You'll also usually enjoy a more lenient response from users since you've explained that the features aren't ready yet.

Of course, you'll also probably get honest and negative feedback. We've found, however, that this kind of feedback usually has much less impact on the overall perception of the product and the brand. The only exception is if your new beta feature is so poorly developed that the main product can't run anymore. Then, you'll hear that quickly and clearly from your users. But then, this issue is mainly a technical issue, not a usability and UX problem.

With the feedback you receive on beta features, you can also enter into a dialog with product managers and provide your assessments and recommendations regarding the features. This feedback, in turn, can be incorporated into the prioritization of implementations and improvements or, if necessary, lead to certain features no longer being developed.

8.3 Plan for the Appropriate Amount of Effort

This section is a crucial part of the book, and we realize that many questions and thoughts, especially at the beginning of a project, revolve around the question, "How much effort do I need to plan for?"

Unfortunately, no simple answer to this question exists, only rules of thumb that can help you decide. As you can imagine, we believe that any time and money spent on usability and UX design are always well invested. However, this idea is true only up to a certain point. We want to provide you with the pragmatic rule sets we usually work by. If you know any other helpful rules, feel free to send them to us! We welcome e-mails to *handbook3DUX@custom-interactions.com*.

When planning the amount of effort, we generally follow these rules:

- **Rule 1: The more inflexible, the more time**
  If you can't change anything about your system, product, or service after the release, or if you would have to invest a lot of effort to do so, you should invest as much time as possible in advance. Of course, we don't mean that you endlessly "UX" away, but you should consider when your solution is good enough and then work on it for just as long until you have reached this goal.
- **Rule 2: The more risk, the more time**
  The riskier your application is, the more time you should invest. But even then, the following often applies: Determine when you have reached your goal. This determination is often the case for critical applications when something specific no longer happens, for instance, when certain tasks can be completed without errors, when training is no longer needed, or when no one else needs to be consulted.
- **Rule 3: The more crucial the application, the more time**
  The more important the designed workflow is for the application, the more time you should invest in optimization. This rule refers to the criticality of a workflow for your application and not the criticality of a workflow for the user.

  For example, imagine you're developing an app. The app is entirely uncritical, and its use does not create a risk for anyone. You then decide not to test the app but to publish it directly. In the app store, however, you suddenly realize you're collecting only 1-star ratings because the logon process does not work correctly. Since logging on decides whether your app can be used, this function is critical for the app. Thus, you should ensure that everything works that is needed to use the primary function of the system, product, or service.
- **Rule 4: The better known, the more time**
  If you're highly well-known with your product or brand and have a (positive) reputation to lose, then you should also plan more time than if you're unknown and new. In this context, it is simply a matter of customers and users associating a certain quality expectation with you. Failing to meet these expectations will reflect negatively on you. Thus, whenever you have something to lose, it pays to invest more time in usability and UX design.
- **Rule 5: The more innovative, the more time**
  If your approach is entirely innovative and how people can use and interact with it is unique in the market, investing more time is worthwhile. You can't yet draw on

empirical data—neither your own nor that of your competitors. In our view, the probability of an entirely new concept being directly successful at the first attempt without user research and testing is extremely low. So far, we've never seen a product that came into contact with users for the first time and did not exhibit any usability or UX problems at any point.

However, if your product has a significant technological added value (you enable something that couldn't be done before), then it might work without superb usability and UX.

- **Rule 6: The more outstanding UX is positioned as a unique selling proposition, the more time**

  The more you use good usability and UX as a unique selling point over competing products, the more time you should invest in usability and UX design. This decision is primarily a product strategy decision that we often encounter in areas that are so technically advanced that the range of functions between different competitor products is almost identical.

  Setting your product apart from its competition can make sense through a remarkably accessible and appealing design. This design can succeed in shortening familiarization, training, and usage times, in this way generating more enthusiasm and thus broader dissemination. In the best case, the product is opened up to new target groups, which makes the competitive advantage particularly clear.

- **Rule 7: The more complex, the more time**

  The more extensive and complex your application is, the more time you should invest in designing good usability and UX. Especially if you're mapping workflows or designing expert systems, many companies follow the assumption that less effort can be invested in usability and UX.

  But when was the last time you hired someone (or when was the last time you were hired) for the ability to operate a particular program, not for what you can accomplish with the program? The more time spent on the operation, the less time is left for the actual task. Of course, this cost is true not only with expert systems but with all complex applications. Just because you could operate a complicated application with many usability and UX weaknesses doesn't mean you want to!

- **Rule 8: Every main workflow and every high-risk workflow should be tested at least twice**

  To ensure that your solution does no harm and performs its primary function, we recommend testing the main workflows and the risky workflows with users in at least two iterations before release.

  By *main workflows*, we mean the processes required to use the core functions of the solution and that—if they did not work or are not understood—would result in the solution no longer delivering value. Committing to two iterations ensures that you have received feedback on the initial idea and have checked at least once whether the path you have taken to improve on the initial design is on target.

Chapter 9

Data-Driven UX Design in Detail: Context of Use Analysis

"Nothing is absolute. Everything changes, everything moves, everything revolves, everything flies and goes away."
—Frida Kahlo (painter)

Now the time has come, dear reader: We start with the first "real" step of our usability and user experience (UX) design! This chapter is about analyzing your product's *context of use*, as shown in Figure 9.1.

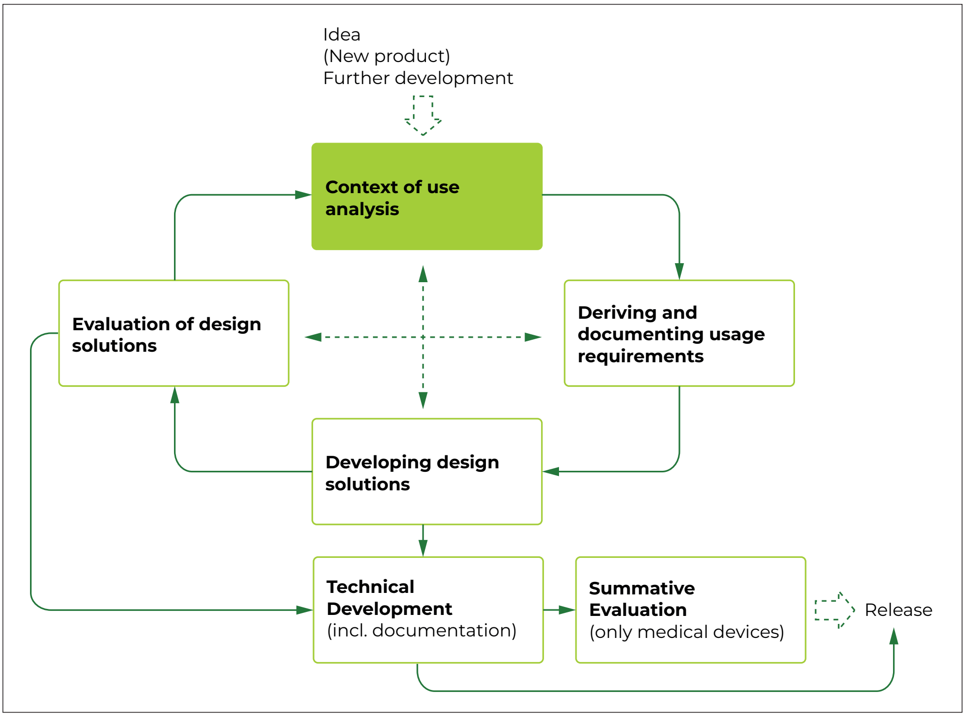

Idea
(New product)
Further development
Context of use analysis
Evaluation of design solutions
Deriving and documenting usage requirements
Developing design solutions
Technical Development
(incl. documentation)
Summative Evaluation
(only medical devices)
Release

Figure 9.1 The data-driven UX design process (3DUX): The context of use analysis.

Recall the definition of "context of use" from Chapter 2, Section 2.2.5, based on ISO 9241-11:

> *"Combination of users, goals and tasks, resources, and environment."*
> *(ISO 9241-11:2018)*

The term *analysis* on the other hand, is described in the *Merriam-Webster Dictionary* as:

> *"A detailed examination of anything complex in order to understand its nature or to determine its essential features." (www.merriam-webster.com)*

The first step of the data-driven UX design process—the context of use analysis—is about looking at the combination of users, goals, tasks, resources, and environment and breaking it down into its components. But why? We'll shed some light on that question in detail in the next section.

9.1 What Questions Should the Context of Use Analysis Answer?

If usability and UX design were a house, then the context of use analysis would be the foundation. The findings and assumptions made in this step often form the basis for the entire development. Incorrect assumptions can cause the whole house to collapse, whereas correct assumptions can form the basis for a successful product.

At its core, the context of use analysis is about understanding the *current situation*. If we follow the earlier definition of context of use, then this analysis is made up of the following elements:

- Users (note that patients in the medical device context take on a unique role depending on the medical device since they do not always operate it themselves, but it is used on them)
- Goals
- Tasks
- Resources
- Environment

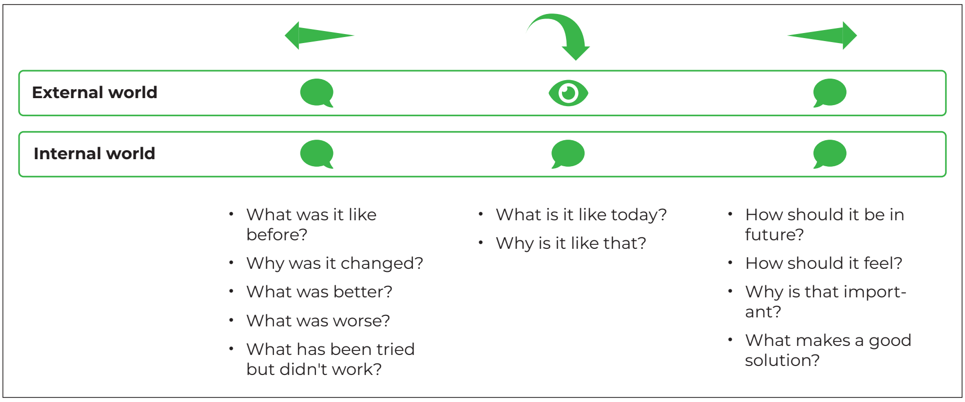
However, our emphasis on the "current situation" does not only refer to the present. In the context of use analysis, we are also interested in how this situation came about.

To understand this statement more quickly, consider our discussion from Chapter 6, Section 6.3.2. We distinguished the dimensions of questioning and observation in the inner and outer world according to the past, present, and future, as shown in Figure 9.2.

Let's focus on this systematization of questions. If we were to limit the context of use analysis to the *present* and analyze the actual situation, we would answer questions such as the following:

- For whom is the product intended?
- In the medical device sector, for which patient group is the product intended?
- What are the goals the users are pursuing?
- What are their tasks (with the product)?

- What resources are available to them?
- In what environment(s) do they operate?



- What was it like before?
- Why was it changed?
- What was better?
- What was worse?
- What has been tried but didn't work?

- What is it like today?
- Why is it like that?

- How should it be in future?
- How should it feel?
- Why is that important?
- What makes a good solution?

Figure 9.2 Question structure according to the external and internal world and the past, present, and future. This structure also shows what is observable and, thus, what possibilities questioning offers.

These questions are important, and we'll also present methods later in this chapter with which you can answer precisely these questions for yourself and your project. Nevertheless, we call upon you to consider the *past* as part of the actual situation. By emphasizing the past, we do not mean having an historical view but rather a retrospective view of all the individual points of the context of use. Thus, we ask a series of complementary questions, such as the following:

- Who is the product intended for? Who was it intended for in the past? How did the change come about?
- In the medical device sector, for which patient group is the product intended? For which patient group was the product previously intended? How did the patient group change?
- What are the goals of the users? Were there other goals in the past? How did their goals change?
- What tasks do the users have (with the product)? How have the tasks changed compared to before? How did the change come about?
- What resources are available to the users? Have these resources changed compared to the past? How did the change come about?
- In which environment(s) do users operate? How has the environment changed compared to the past? How did the change come about?

We hope you recognize what we are getting at. Our goal in the context of use analysis is to understand the current state, but in such a holistic way that we can also directly exclude solutions that have already been tried and did not work in this context of use.

In most cases, you'll find that users and manufacturers are often rather creative and that many "obvious" solutions, which directly catch your eye as an "external" usability and UX designer, have not yet found their way into the everyday life of users for certain (and mostly understandable) reasons.

But now, let's look at the context of use analysis in detail.

9.2 What Is the Context of Use Anyway?

As mentioned earlier, the context of use consists of many different areas. We have already introduced the individual areas for you, but in this section, we want to take a detailed look at each component of the context of use. We'll start with the users—how could it be any different in data-driven UX design?

9.2.1 The Users

Recall our earlier definition of users from Chapter 2, Section 2.2.4, based on ISO 9241-11: *Users are the persons who interact with a system, product, or service*.

We take up this definition again because users represent a central element in the context of use analysis and the entire data-driven UX design process. As described earlier, an essential task is to understand that this includes not only the people who use the system, product, or service. This term also includes individuals who use the system, product, or service to perform the following tasks:

- Operate the solution
- Use the results
- Support operations (including maintenance and training)

All these people are users in one way or another and perform specific tasks with the system, product, or service. Since you, as the designer of the solution, must enable users to perform their tasks, you must get to know them and be able to describe them. The characteristics that are important in describing users in this context are directly related to the solution to be designed. Typical characteristics that are considered in the context of a use analysis include the following:

- Job title (in the work context)
- Work experience (in the work context)
- Age
- Education level
- Further training
- Special abilities (e.g., special knowledge or language skills)
- Special skills (e.g., trained processes in the trade)

- Special physical characteristics (e.g., in air traffic control, only persons with perfect eyesight—or, to boast, with "perfect visual acuity"—are allowed to work)
- Special psychological characteristics of the user (e.g., a particular limitation or special aptitude)
- A special condition of the user at the time of use (e.g., something is only used at night/fatigue)

That's quite a few characteristics. Two disclaimers directly to this:

1. Not all of these points will be relevant in your case.
2. Other points may be included that are missing here but are particularly relevant to *your project*.

Therefore, before you go off and collect a massive list of criteria, take the time to consider which criteria are relevant to you and your system, product, or service.

For example, suppose you conduct business purely in the private sphere. In that case, neither job title nor special training will likely influence your solution, design, or business model. Equally unlikely to be relevant to an application sold to businesses is a person's income. As always, exceptions to these rules exist.

For us, a rule has proven useful only to record and systematically collect important information for the situation. In practice, not only is lacking information bad, so too is an overabundance of information from which you'll need to filter out the critical information. Have you ever stood in the supermarket and had to choose between 47 types of toilet paper? If so, you know exactly what we mean by "information overload."

In this context, of course, your experience in user research comes into play. But even without much experience, you'll get a feel about whether a piece of information seems essential. The good thing about data-driven UX design is that the process is an iterative approach. So, you'll go through the context of use analysis several times and add to it or adjust it. Therefore, our tip is to not dwell too long on making a "complete" list.

9.2.2 The Patients (Medical Devices)

Since one example comes from medical devices, and usability plays a unique role here, we would also like to address another particular group: patients. In itself, you can think of patients the same way as users and largely use the same dimensions to describe patients.

Patients, however, represent a particular case for us since—depending on the medical device—they are not always direct users of a product but are directly affected by it. Of course, this case also exists outside the medical device industry (e.g., passenger management systems at airports). However, the effects are usually much more severe in the case of medical devices, which is why we have decided against considering patients as part of the usage environment or simply as additional stakeholders. Instead, we'll list this group directly and explicitly as a separate part of the context of use analysis.

If you're concerned about patients, then from our point of view, you should additionally ask yourself the following questions:

- Is the patient also the product user?
- Is the patient systematically limited in any particular way by their disease?
- Is the patient vulnerable in a particular way?

Let's work through these individual questions in detail.

Is the Patient Also the User of the Product?

This question is about understanding if this patient will operate the product themselves. If yes, you can use your user description and add the other questions. An example of such a product is a blood pressure monitor for home use, as shown in Figure 9.3.

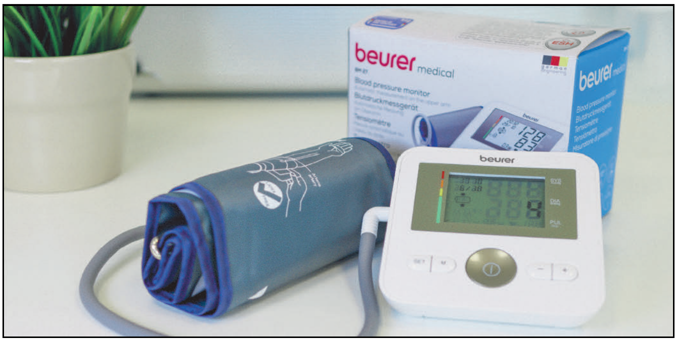


Figure 9.3 Beurer blood pressure monitor for use at home.

If the user and patient are not identical, creating an additional description of the patient and answering all questions from the patient's point of view makes sense. We'll introduce you to the best way to document this knowledge later when we discuss personas in Section 9.4.3.

Is the Patient Systematically Limited by Their Disease?

Regardless of whether the user and the patient are identical, a best practice is to consider whether the patient is limited in a particular way by the disease (or associated characteristics). Let's illustrate this question through an example.

Imagine you're developing an insulin pump for diabetics to use. Although doctors set up the pump itself, the device is attached to the patient and used by the patient daily. In this case, several users are involved, each with different tasks. One user is "additionally" the patient. Thus, considering both physician and patient individually in the context of use analysis is important. Regarding the limitation of the patient due to the disease, in this case, patients with insulin pumps are, in most cases, type 1 diabetics. For these patients, reduced sensitivity or slow healing from injury (because of poorly adjusted sugar levels) can be symptomatic.

In addition, type 1 diabetes is an autoimmune disorder and often occurs at a young age (in children and adolescents), which also results in additional requirements for your product, even if these requirements do not directly affect the disease itself. Because some patients may be children, you must assume that their motor skills (e.g., their dexterity and strength) are lower than adults and that children cannot read or can only read poorly. Thus, the opportunities you have to guide your younger patients differ significantly from the options you have with adult patients.

You must assess whether these special requirements are essential for your product on a case-by-case basis. However, you can only do this assessment successfully if you know what the special requirements are. Especially in the field of medical devices, you should work for an overall picture beforehand so you can decide what is important and what is not.

Is the Patient Vulnerable?

The last question concerns the need for protection. Of course, users outside the medical field can also require protection, so you're also welcome to include this question in your general process. For example, we already addressed children as users in the previous section. They are often considered a particularly vulnerable group.

Back to patients and medical devices: Due to the unique nature of medical devices, primarily used for diagnosing or treating diseases, medical devices are commonly used on persons with fewer options to protect themselves independently.

Your recognition of this need for protection and its consideration in your subsequent design would be greatly appreciated.

9.2.3 The Task

Following ISO 9241-11:2018, a *task* can be defined in the following way:

> *"Set of activities undertaken in order to achieve a specific goal."*

A *task* and a *goal* are, therefore, directly related and interconnected. A task is the step-by-step implementation of a goal.

Understanding a user's tasks and goals is a core aspect of the context of use analysis. Sometimes, however, identifying clear tasks and goals is difficult for the following reasons:

1. Often, users have *multiple tasks and goals*, so it is sometimes difficult to understand the boundaries, transitions, commonalities, or backgrounds of the individual task processing from the outside.

2. The tasks are characterized in their execution by the *tools* available to the users for this purpose. Consider, for example, a telephone book. When using a telephone book, the task is not to search for a specific number. The goal is actually to contact a specific person. Searching the phone book only becomes necessary because that is how the current tool works. So, separating the task from the tool is a challenge. In this context, you can use the task or task analysis (Section 9.4) to distinguish between tasks and individual action steps and thus become more solution neutral.
3. Parts of the task are not visible because they may be *perceptual tasks* (e.g., reading a characteristic value) or *cognitive tasks* (e.g., making a decision) that are nevertheless core elements of a task. In this context, an essential step is to gain an overall understanding of the task so that a redesign can be implemented successfully.

Tasks, however, do not usually stand in isolation but generally have a *trigger*. This trigger is usually located in the person's goals, which are also part of the context of use, which we'll look at in the next section.

9.2.4 The Goals

According to ISO 9241-11:2018, a *goal* is an "intended outcome," which fits well with the idea of progress according to the *jobs theory* from by Clayton M. Christensen and colleagues (2017). This theory assumes that products that enter a person's life do so because they are "tasked" with enabling specific progress. Christensen defines a goal or (as he puts it) a "job" in the following way:

> *"The progress a person wants to make in a particular circumstance."*

According to jobs theory, goals often exist on different levels (see Chapter 6, Section 6.2.1, where we introduced you to jobs theory). This is relevant concerning the context of use analysis since we want to identify user and stakeholder goals. Let's, therefore, take a closer look at the concept of *levels according to the jobs theory*.

The Levels of a Goal

When we speak of goals, most people have in mind that a goal is something you have to do or achieve and that this goal is visible and observable in some way. In fact, these characteristics are true for most goals.

Nevertheless, thinking only about these functional goals of the user is insufficient when you trying to develop a successful and well-designed product. In our experience, very few products fail because they don't serve the functional level meaningfully but instead because they flub one of the other levels. The goals with which users (and customers) commission a product can involve one of the following three levels:

- The functional level
- The social level
- The emotional level

Since we have already introduced these levels in detail in Chapter 6, Section 6.2.1, we'll only summarize what you need to understand in this section: The *functional level* describes what needs to be done. On the other hand, the *social level* is about how you want to be perceived and how you interact with others. Looking at the *emotional level*, the focus is on how we want to feel.

A Striking Example

Let's consider a rather striking example in this section to clarify the different levels of a goal and the resulting challenges for context of use analysis. Imagine you're married, and your wedding anniversary is coming up in a few days. Unfortunately, you don't have a gift yet.

If we were focused solely on the functional level of your goal (getting a gift), you could march into the nearest store and get some socks for your partner. However, you'll probably agree that this gift would not be the best gift for your wedding anniversary due to your job's social and emotional level (getting a gift).

After all, you don't just need any gift. You need a gift that will show your partner that you know them very well, really appreciate them, and have put a lot of thought into it (social level). Additionally, with this incredibly thoughtful gift, you probably want to make sure your partner is pleased and excited about the gift, giving you the security and relaxation you need to approach your next anniversary with peace of mind (emotional level).

Now, if you were asked by a third party what you were doing, you would probably say, "I'm picking out a wedding gift." Thus, fully capturing the goal within the context of use analysis is not easy because a certain vulnerability comes with disclosing one's personal goals.

However, this example illustrates why some solutions seem to fit well from the outside but are vehemently rejected. Experience shows that these solutions then support the achievement of the functional goal, which is openly discussed, but miss goals on the social and emotional levels. Conflicts are more challenging to uncover and harder to resolve because the actual background is not readily apparent. However, uncovering them is the basis for developing meaningful solutions.

9.2.5 The Resources

Another aspect that should be considered for the context of use analysis is the analysis of existing resources that users have at their disposal. Regarding the context of use analysis, we can divide resources into *hardware*, *software*, *material*, and *information*. Note that this division into different categories is not an exact science and is only intended to be a heuristic aid.

In the context of use analysis, or more precisely in analyzing the resources available to a user, the categories hardware, software, and material are usually straightforward to identify because they are visible. According to the approach introduced in Chapter 6, Section 6.3.1, these resources are located in the *external world* of the user. The information category, however, is often only indirectly identifiable because information generally resides in the user's *inner world*.

Let's go through each resource category in detail with some examples.

Hardware

By hardware, we mean all work equipment that is used with a direct relationship to the task or the user's goals.

Let's say, working on the moveHUB team, we're doing a context of use analysis at a train station. In this case, the timetable displays would be a form of hardware. In a broader sense, ticket vending machines could also be a form of hardware, although software runs on this hardware. Fuzzy classification may thus be required. We could classify the ticket vending machine as hardware, software, or both. In practice, we focus on the task in question.

For moveHUB, we would classify the ticket vending machine as hardware since the software is predefined for the users. The software is part of the device and is permanently connected to it, at least from the user perspective. In addition, users cannot do anything else with the ticketing software.

If the users could install other software or perform other tasks, we would put this part in the software category instead. In the case of a PC with various applications, for example, we would include the PC as hardware, and its various applications would fall into the software category.

Software

Generally speaking, software includes all types of applications used for a task or to achieve the goals. Whether the software subsequently runs on a PC, tablet, smartphone, smartwatch, or smart glasses is irrelevant. However, we would list the necessary hardware in the hardware section, with a link to the software used.

Again, this discussion of software is open to debate, and you'll get the same answer from us: The point is not that you can provide rock-solid divisions into various categories but that you know the map into which your product will have to fit later. If you find out that other categories or divisions work better for your project, feel free to adopt them.

Material

In addition to hardware and software, we also consider the material area. Under *material*, we include all those things necessary for the task and goals that are not used permanently but instead once or only a few times before being replaced with new things. Typical items might be paper and pens or, in the medical field, gloves and bandages.

Information

In our opinion, the final significant resource is information. Information can be provided in various ways and is at least partially included in the other categories. For example, hardware products (such as timetable displays) or software products (such as a patient management program) will mainly provide information. However, in addition to the information already documented, some information may be delivered verbally and still be critical to the task and goals.

For example, in a context of use analysis for an ATM, you would find that, apart from the hardware and software (ATM/cash card) and the material (a remittance slip or even bank bills), something else plays a role: the secret PIN number. Without this information, task processing is not possible.

However, we can list information separately also because the knowledge of the required information allows you to create even wholly new and different solutions.

In the moveHUB example, a timetable display depends not on the hardware but on the information it provides. If you come up with an idea to provide the same information in another, easier way, then you may have found a new product idea.

The telephone book is an example from the private sphere where this change process is probably pretty straightforward. When we were younger, it was an important event of the year to stop by the local post office in time to grab one of the phone books. Today, to be honest, can you even remember the last time you were at the local post office? We don't even have to talk about using a telephone book. First, the phone book was replaced by an online version like *whitepages.com*, but now fewer and fewer people even have an entry there, so the information itself is losing value. Phone books, both physical and digital, have been replaced by *social networks* like Facebook (almost no longer), Instagram, or LinkedIn, which help maintain or establish contact in other ways. In the business world, websites like *yellowpages.com* may still exist, but most people use search engines like Google to find and call a business.

Therefore, we hope you take away the realization that the information needed for a task and a goal is often the same, but how this information is implemented or provided can change significantly over time. From our perspective, identifying the information needed for a task is the core element in determining resources.

9.2.6 The Environment

The final element of the context of use is the environment. The environment can again be subdivided into the *physical*, *social*, *organizational*, or *regulatory environment*.

Physical Environment

The term *physical environment* includes, on the one hand, aspects such as dimensions of rooms and distances between objects as well as issues like temperature, humidity, lighting, and ambient noise level. In addition, a distinction should be made as to whether a product will be used indoors or outdoors. These decisions significantly impact the choice of technology, usability, and UX design. For example, if the product is used outdoors, operation might need to be possible in gloves or even in heavy rain.

Social Environment

The question of whether a product is used indoors or outdoors is also indirectly related to the social environment of the product. The *social environment* describes whether a product is used alone or together with other people and whether the people with whom it is used together, if applicable, are acquaintances or strangers. This shared action can involve both *co-use* (i.e., actual use of the product together with several people) and simply the use of the product in the presence of others.

When using a product in the presence of others, additional topics such as privacy or secrecy suddenly become relevant. These issues would not be clear or important in the case of a ticket vending machine, but suddenly are all the more critical when our case involves an ATM. Even though the ATM is placed in a public space, you must ensure that every user can interact securely and privately without enabling every bystander to read their PIN number.

The Organizational or Regulatory Environment

In addition, we often also consider the *organizational or regulatory environment* of users. The *regulatory environment* is the totality of all applicable rules that users must observe. These regulations can include laws, standards, or corporate processes. For example, in a private environment, the regulatory aspect is usually negligible. In a professional environment, however, both the organization environment and the regulatory environment often play a decisive role.

To consider the organizational and regulatory environment, you should ask yourself the following questions in particular:

- What is allowed?
- What is desired?
- What is undesirable?
- What is forbidden?

Knowing the *organizational and regulatory environment* sets the framework for solving user problems, setting new ways of doing things, and achieving goals. Anything outside the organizational and regulatory framework is conceivable but requires a much more extended implementation period and may have higher hurdles.

One good example is the recent introduction of autonomous vehicles. Work on research and development of (partially) autonomous vehicles began 15 years ago. At that time, however, prototypes were not yet allowed to drive on public roads, which limited the possibilities for the solution itself as well as for solution development. At the time, we made do with closed test tracks. Automakers also relied on test tracks and created testing facilities in California, where driving on the road was (and is) legally allowed.

As you can see, limitations in the organizational or regulatory environment should not necessarily mean inaction. However, you should consciously choose this path beforehand and know you'll need staying power. You'll get this knowledge from the context of use analysis.

9.2.7 Risk, Frequency, and Conclusion

Context of use analysis is fundamentally essential for subsequent data-driven UX design process activities. At the same time, this analysis is also multilayered and often complex. If you do an excellent job in this step, however, the subsequent steps will be much easier.

In this context, we would like to mention one more point also discussed in Chapter 5, Section 5.2.2: *risk consideration*. Vitally, the context of use analysis helps you understand precisely which operating tasks have the highest risk for errors in usage.

In addition, you can understand the *frequencies of individual tasks* for user practice. Again, this information will help you guide the next steps of data-driven UX design. As mentioned elsewhere (for example, in Chapter 5, Section 5.2.4), you should first address high-risk operations. Next, you can design operating procedures that are not associated with high risk but will occur frequently. Finally, you can address the "remaining" operations (low-risk and rarely used functions).

In the end, the data from the context of use analysis helps you twice. You can develop a better product and manage the development itself more efficiently. In the next section, let's turn to how the context of use analysis can be performed.

9.3 Methods of Context of Use Analysis

You now know what the context of use is and how it is composed. You also know how important the detailed recording of the context of use is for your product development. We hope you have also concluded that you do not need all the information but must constantly decide what is important for your system, product, or service. But what remains open is how you get all this information. To this end, in this section, we'll introduce you to various methods with which you can systematically investigate the context of use and document the results:

- Secondary research/desk research
- Field observation
- Interview/jobs-to-be-done
- Group discussions/focus groups
- Quantitative survey/questionnaires
- Contextual inquiry
- Analysis of quantitative usage data
- Web tracking/click analysis
- Ticket and feedback analysis

We'll also introduce you to methods for preparing and presenting the obtained data in an understandable way in Section 9.4, such as the following methods:

- Task analysis
- User or customer journey mapping/touchpoint analysis
- Personas and empathy maps

Let's start with secondary research and desk research.

9.3.1 Secondary Research/Desk Research

The statistics website *Statista* (*www.statista.com*) predicts that the volume of digital data generated annually worldwide will increase from 33 zettabytes in 2018 to 2,142 zettabytes in 2035 (Statista 2023). To be honest: We can't even make sense of this number (with its 21 zeros). But what this extremely large number is good for is the following:

- We want to highlight that the amount of available information increases enormously every year and that it has never been easier to get (good) information at a low cost.
- We want to give you a direct example of what secondary research can look like.

We recommend starting any data-driven UX design project with some secondary research. This step will give you a quick overview of the existing knowledge and allow you to uncover knowledge gaps more quickly, which you can then close by applying other methods in your context of use analysis.

Secondary research is not a fixed method that always follows the same procedure but refers to whether you conduct surveys, observations, group discussions, or similar yourself, or instead draw on data created by others. Depending on the environment in which you operate, this data could already exist within your company or might be data provided by external parties. Examples of typical internal company data could include the following:

- Data on turnover or sales statistics
- Information from cost accounting
- Results from previously conducted market research

From this information, you can deduce which of your existing products are particularly popular, how the popularity of the products has changed over time, which customer groups you have, and how these have changed over time. But you can also gain insights into your product's profitability or what competitive advantages your customers think you have over your competitors. The more data your company has and the better it is prepared, the smaller and sharper the knowledge gap you'll have to fill afterward.

Other external data sources can be used for secondary research for all data not available internally in a company. An important building block is statistical data provided by federal statistical agencies, for example, or collected by market and opinion researchers and made public in regular reports. Statistical data can help you better understand market sizes, total sales, or social trends.

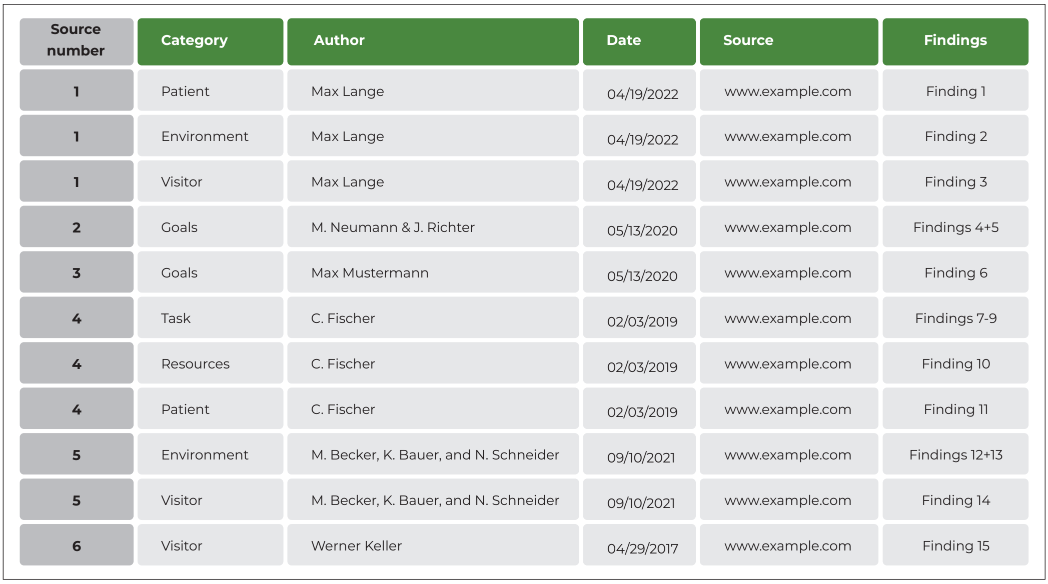
In addition, interest groups, professional associations, insurers, and public institutions can be used as contact points. These players usually have extensive data on their members and regularly publish these findings in industry reports or white papers.

Depending on the subject area in which you're working, however, it may also be helpful to search for patents or new technologies, or you could check the content of presentations at suitable specialist congresses. Just as necessary could be journals and newsletters, doctoral theses and other final papers, or reference books. The list of possible sources of information is almost endless. And if *Statista* is to be believed, these sources continue to multiply daily, which leads us directly to the next challenge: data preparation from secondary research.

The data gathered from secondary research can be prepared in different ways. How you prepare and further use this data again depends on your objective. In any case, the first sensible step is to prepare a research protocol, as shown in Figure 9.4, in which you document—for example, in the form of a digital table—the location, the author, the date, and the core findings of the source.

To rearrange the table as you like but still have a unique reference, in our experience, it makes sense to add a source number as well. In this way, you avoid forgetting any sources and avoid researching any content twice.

However, the search log is only suitable to a limited extent for obtaining the overall picture. A first step towards creating this picture is to add a "Category" column to the search log to enable sorting. In this column, for example, you could use the individual areas of the context of use (user, patient, task, goals, resources, and environment), in the best case, of course, already neatly divided into subgroups.



Source number	Category	Author	Date	Source	Findings
1	Patient	Max Lange	04/19/2022	www.example.com	Finding 1
1	Environment	Max Lange	04/19/2022	www.example.com	Finding 2
1	Visitor	Max Lange	04/19/2022	www.example.com	Finding 3
2	Goals	M. Neumann & J. Richter	05/13/2020	www.example.com	Findings 4+5
3	Goals	Max Mustermann	05/13/2020	www.example.com	Finding 6
4	Task	C. Fischer	02/03/2019	www.example.com	Findings 7-9
4	Resources	C. Fischer	02/03/2019	www.example.com	Finding 10
4	Patient	C. Fischer	02/03/2019	www.example.com	Finding 11
5	Environment	M. Becker, K. Bauer, and N. Schneider	09/10/2021	www.example.com	Findings 12+13
5	Visitor	M. Becker, K. Bauer, and N. Schneider	09/10/2021	www.example.com	Finding 14
6	Visitor	Werner Keller	04/29/2017	www.example.com	Finding 15

Figure 9.4 Simple research protocol for secondary research.

If you want to work more visually, creating a mind map with important content based on the pre-grouping can be helpful. A mind map is a graphical representation method that shows how the different content belong together, thus greatly simplifying the discussion with other people (e.g., stakeholders). An example mind map for the context of use is shown in Figure 9.5.

To keep the complexity of the mind map as low as possible and still be able to check later whether all relevant information is contained in the mind map, we recommend that you label each entry in the mind map with the source number from the table. If you have any questions or uncertainties, you can easily find the source of the information and, if necessary, re-read it if any ambiguities arise.

As soon as you have this mind map in front of you, what should catch your eye are points that have little to no information from secondary research. You'll recognize this lack of information because—if you follow the structure of the context of use with the mind map —some individual nodes in the mind map may only have a few sources. This imbalance gives you a good visual overview of the current status of your understanding, and you can start looking specifically for information to supplement deficiencies in the mind map.

Thus, in our view, secondary research pursues several goals in parallel, namely, the following:

1. Secondary research can help you identify unanswered questions and, therefore, determine where it is essential to conduct studies.

2. You get to know the problem area better and where you are with your solution.
3. You may even form new assumptions or reinforce or discard your existing assumptions.

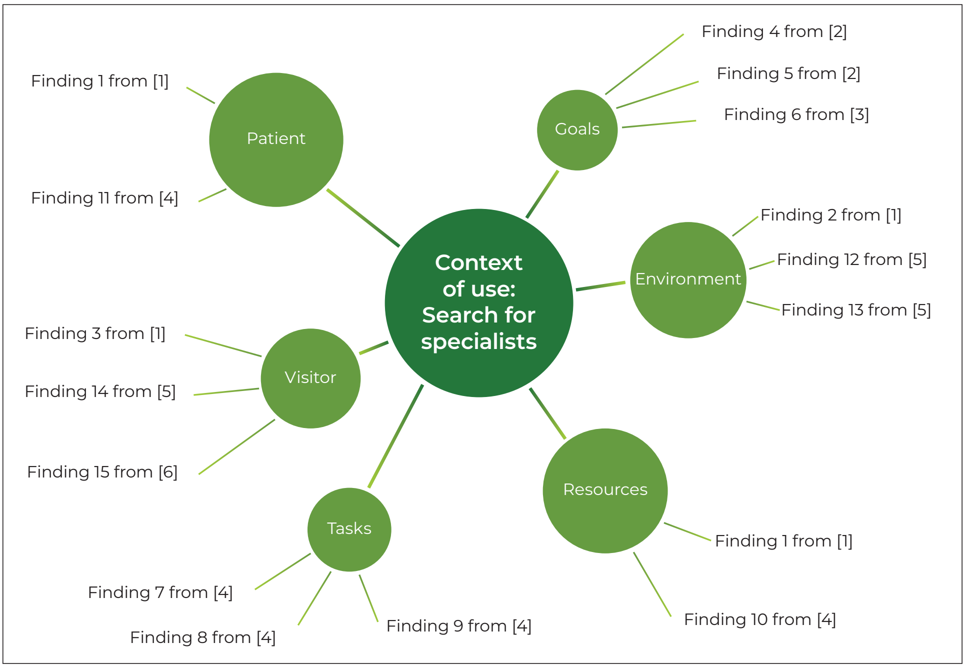

Finding 4 from [2]
Finding 1 from [1]
Finding 5 from [2]
Patient
Goals
Finding 6 from [3]
Finding 11 from [4]
Finding 2 from [1]
Context of use: Search for specialists
Finding 12 from [5]
Environment
Finding 13 from [5]
Finding 3 from [1]
Visitor
Finding 14 from [5]
Finding 15 from [6]
Resources
Tasks
Finding 1 from [1]
Finding 7 from [4]
Finding 9 from [4]
Finding 10 from [4]
Finding 8 from [4]

Figure 9.5 Mind map for preparing research results from secondary research.

Advantages and Disadvantages

The advantages of secondary research are clear:

- Using existing data makes familiarization with a topic much more accessible.
- Costs are reduced because studies do not have to be carried out by the company itself.
- Certainty in decision-making is increased because more knowledge is available.

Unfortunately, the disadvantages of secondary research are as evident as the advantages:

- The available data is often general or does not precisely answer our questions.
- Data quality can only be surveyed indirectly because the raw data is usually not accessible, and only interpreted data is available.
- The data is also available to anyone else, so owning the knowledge is not a competitive advantage. However, the interpretation of the data is solely in your hands.
- Data from secondary research may be out of date.

Tips for Dealing with Secondary Research

When working with secondary research, a few essential points should be considered when selecting sources, such as the following:

- Check the timeliness of the data. It usually takes one to two years before studies are published in scientific journals. Therefore, consider whether the data still fits now.
- Check whether the research question is congruent with your research question so that the results can be transferred. Also, pay attention to the context in which the data was obtained. The context can include the people, culture, organizational, and legal framework.
- Check whether the methodology used is suitable for actually making statements. We recommend distinguishing between exploratory studies, in which correlations are discovered by chance, and studies that specifically test suspected correlations. The latter usually has a higher significance.
- When selecting studies and literature, look for what are called *peer-reviewed journals*. These journals are committed to certain academic quality standards. To meet this standard, other scientists read the articles before publication, thus assuring that several experts have considered the methodology and evaluation valuable.

Table 9.1 briefly describes this method and its advantages and disadvantages in the form of a "profile."

Profile: Secondary Research	
Brief description	Secondary research is the use of existing material, existing information, and previously conducted studies to answer a research question of one's own.
Advantages	■ Using existing data makes familiarization with a topic much more accessible. ■ Costs are reduced because studies aren't carried out by the company itself. ■ Certainty in decision-making is increased because more knowledge is available.
Disadvantages	■ The available data is often general or does not precisely answer our questions. ■ Data quality can only be surveyed indirectly because the raw data is usually not accessible, and only interpreted data is available. ■ The data is also available to everyone else, so possessing the knowledge is not a competitive advantage. ■ Data from secondary research may be out of date.

Table 9.1 Profile of the secondary research method.

9.3.2 Field Observation

Primary research must include everything that does not come from secondary research. Field observation is one of the most basic methods for obtaining data for a context of use analysis.

Field observation is about going on-site in the usage environment where the product, system, or service will later be used or—if that environment is still unclear—to the place where the problem you want to solve can be observed.

In the field observation method, the name says it all. To execute the method flawlessly "by the book," you would not interact with people you are observing but remain a completely passive observer. Whether you carry out this observation directly (personally on-site) or indirectly (via video), openly (the observed persons know that they are being observed) or covertly (the observed persons do *not* know that they are being observed), depends again strongly on the environment and the legal situation.

In general, field observation can focus on different aspects (as described in Chapter 6, Section 6.3.1):

1. You can observe a person
2. You can observe a process.
3. You can observe the environment.

In most cases, you'll observe a mixture of these kinds of content during a field observation. Depending on your research question, however, you'll set a different focus.

For example, imagine you're one of moveHUB's founders, and your goal is to give people from rural areas who regularly commute to cities a better overview of available transportation options. Since you have a particular target group, your first field observation could focus on observing people commuting. This would put the focus on commuters themselves. To conduct a field observation, our founders Matthew, Emma, and Anthony go to a local train station, each following a person taking the train to the nearest city.

BusinessBooster, on the other hand, is about managing contacts and sales processes. Several people and departments are usually involved in the initiation of offers and the processing of sales. Therefore, observing a single person would not be sufficient in this case. Therefore, the scope of the field observation in this case would be across personnel and perhaps include service employees, consultants, and the back office. The focus in this case is thus on observing the process. In the field observation, Markus, for example, could observe what happens when one of his customers receives a call, how the caller's contact data is recorded, how an offer is calculated, and how the offer is created and transmitted.

If the focus is on the environment—for example, if the activities conducted in a doctor's office must be recorded in detail for our Healthyfy service—then you would refer to this environment in the field observation. Your observation focus would then be on what is happening in the doctor's office.

Put simply, the focus of the field observation describes who you're following, as shown in Figure 9.6:

1. When observing a particular person, you "follow" that person—sometimes, this approach is called *shadowing*.
2. When observing a process, you "track" the task across multiple people, systems, and handoffs.
3. When observing an environment, you're usually static and stay in place without moving away. This observation type is sometimes called the *fly-on-the-wall method* (e.g., Dark Horse Innovation, 2017).

Figure 9.6 Three different field observations with a different focus. Are you observing the environment or following a person or a process?

The evaluation of a field observation can be done afterward or directly during the observation, depending on whether you're allowed to take videos or photos. To make the evaluation as efficient as possible, we recommend the following procedure:

1. Determine your observation's focus (i.e., person, process, or environment) based on your research objectives.
2. Determine what you want to observe (i.e., which people, process, or environment).
3. If you already have an idea, think about categories in advance that you can use to document your observations—in a doctor's office, for example, patient reception, answering the phone, patient medical history gathering, physical examinations, breaks. Caution: Defining categories bears the risk of neglecting everything that does not fit into these categories.
4. Decide what information you need about each activity (e.g., frequency, duration, time, person performing, sequence).

5. Make sure you have space for notes—no matter how much thought you've put into field observation in advance, you'll never be able to anticipate all behaviors. Otherwise, you wouldn't need field observation.

In the simplest form, you have this information prepared as a table and are on-site armed with paper and pen. The live documentation of the data has the advantage that you have an overview of the contents immediately after executing the field observation and can start the evaluation more quickly. However, the disadvantage is that you cannot look at things again at your leisure or check once more whether you have logged the processes correctly. Thus, errors can arise, especially in hectic environments or with complex processes.

Especially if you have little experience in field observation or want to design for a complex activity, we advise you—as far as it is legally permissible—to document it on video.

The *field observation* method itself does not contain any specifications for evaluation. It only describes where and how you can obtain actual data: by observing people, processes, or environments in the actual application context. A typical method of evaluating the data from a field observation is *task analysis*, in which the entire user task is broken down into individual sub-steps, which are then examined individually.

We'll discuss the task analysis in Section 9.4 in detail. In general, however, you must look at the complete data again to document the process correctly. This makes the use of field observations very costly and the analysis extensive.

Advantages and Disadvantages

The advantages of field observation are many and great, with the most important the following:

- You can see how the processes actually operate.
- You'll receive information on aspects that people are not aware of but still have an impact on actions.
- You get uninterpreted/objective data.
- You get information about the context of use.

With these advantages, field observation is a compelling way to gather information about the context of use. Unfortunately, using field observation also has some disadvantages that you should consider in advance, such as the following:

- Some contexts of use are difficult to observe due to access issues (in a private environment, on company premises).
- If you conduct a video-based field observation, you must address the privacy issue in advance.

- Data collection and analysis can become very time-consuming if the observation is to remain objective and is not well prepared.
- You do not get any background information about the behavior of the persons because only observation is done. (For example, the motives of the persons remain mostly hidden because they are not directly observable.)
- Direct field observation (you're on-site) is difficult to scale due to the high effort involved, often resulting in low observation numbers, which makes the data less generalizable and less transferable.

Table 9.2 shows the profile for this method.

Profile: Field Observation	
Brief description	Observation of people, processes, and environments in a real-world usage environment to obtain data about how solutions are used or how specific problems and tasks are handled.
Advantages	■ You can see how the processes actually are. ■ You'll receive information on aspects that people are not aware of but still have an impact on their actions. ■ You'll receive non-interpretive/objective data. ■ You'll receive information about the context of use.
Disadvantages	■ Some contexts of use are difficult to observe due to access issues (in a private environment, on company premises). ■ If you're doing video-based field observations, you must address the privacy issue in advance. ■ Data collection and analysis is very time-consuming if it is to remain objective. ■ You won't get any background on the behavior of the individuals.

Table 9.2 Profile of the field observation method.

Tips in Dealing with Field Observations

- Pay attention to the legal framework when conducting field observations. Since you may be recording video of people, you must familiarize yourself with data protection. In particular, problems can arise in the operational context, as a video recording can also be used for performance appraisals. Especially in larger companies, we recommend that you coordinate the planned field observations with the labor union for this reason, if there is one.
- The evaluation of field observations can be very time-consuming if you do not structure the observation well in advance. You can make your work much easier by considering which situations could occur in the field observation and entering them into your observation sheet. Thus, you only have to "check off" what happened instead of writing down text descriptions.

- If you plan to show the sequence of the task later, we recommend using a system of documentation that makes the sequences of the activities easily visible. For example, instead of just noting the frequency of each activity with checkmarks, you could use numbers to indicate their sequence.

9.3.3 Interview/Jobs-to-Be-Done

The opposite of field observations is interviewing. Rather than passively observing and not interacting with participants, the idea behind interviews is to engage in direct exchange with users and learn more about the user, their tasks, and goals by asking questions, as shown in Figure 9.7.

Figure 9.7 Interview situation.

Looking at the literature, you'll quickly find different forms of interviews (Renner & Jacob, 2020). Probably the most frequently mentioned forms of interviews include the following:

- Structured interview
- Semi-structured interview
- Non-structured interview

When we talk about *interviews* in this section, we always mean conducting *qualitative interviews* that are open-ended and semi-structured, where the evaluation in the field of usability and UX design is often a mixture of analytical-interpretative and quantifying tasks.

In other words, qualitative interviews aim to generate data you can interpret and learn from. Qualitative interviews are thus often exploratory to discover new knowledge, better understand a topic, and allow you to think and empathize with the problem. This task allows you to develop assumptions and formulate hypotheses that you can subsequently test.

According to Berekoven et al. (2009), the central area of application for qualitative interviews is where adaptation to individuality or establishing a relationship of trust is necessary to improve the chances of addressing potentially semi-conscious or sensitive problems through the adapted formulation and sequence of questions. In contrast to observational methods, the content of interviews is primarily concerned with identifying the participant's motives, attitudes, and motivations—in other words, all those aspects where trust is essential for disclosure and that are not directly observable.

Planning the Interviews: The Structure of the Interview Guide

When planning interviews, we recommend thinking, "What do I want to have learned when the interviews are over?" Based on this question, you'll create your interview guide.

In our company, a survey guide is structured based on two dimensions:

- The level of detail
- The time

You can think of the guide as a funnel, where the questions start off broad and become increasingly narrow. The dimension of *time* can be introduced into the funnel in various ways. Generally, we consider the present, the past, and the future in our questioning. In most cases, we move through the funnel step by step.

Thus, you might start generally with the question of how it is today. Then, you could move on to questions that capture how it used to be in general and then end by asking how it should be in general in the future. After that, slide down the funnel a bit and get more detailed. Start again with questions about the present, move to the past, and continue the interview with questions for the future, as shown in Figure 9.8.

In our opinion, the questions for the future are optional and should generally only take up a small part of the interview time. For the context of use analysis, we advise you to focus on understanding the current situation and the tasks and goals.

This scheme for creating an interview guide can be applied to all interviews you conduct regarding usability and UX. The exact content of the interview guide will depend heavily on where you are in development. We introduce the *interview* method in the context of use and therefore assume for the sample questions that you're just interested in understanding the context of use (i.e., the tasks and goals of your users).

If you use interviews at a later stage of development, you would start with different questions if necessary and already in much more detail. Note that, even if we are more advanced in the data-driven UX design process, we continue to question the basic idea of the system, product, or service again every time we speak to users to ensure the development path is solid.

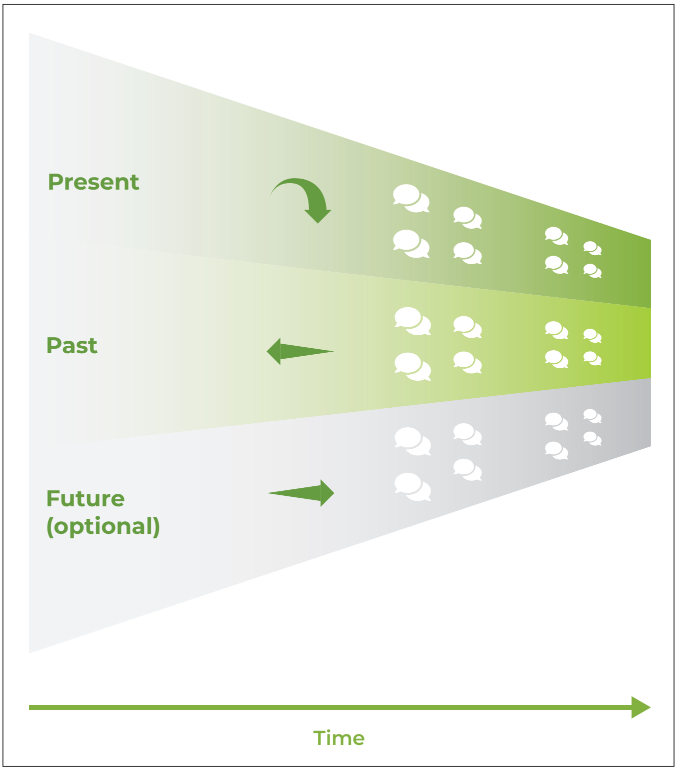


Figure 9.8 Survey guide as a funnel. The deeper you go into the funnel, the more detailed the questions become.

Planning the Interviews: Possible Questions

When creating interview guides for the context of use analysis, we always start with the general, overarching topic. We won't yet ask about future product features or new interface details that will only become important in the upcoming product. To start the interview, we recommend you focus on establishing the current state. We aim to understand whether the problem we want to solve exists for the user and how they deal with it today. Typical questions we might ask include the following:

- What is your current approach to... [insert problem]?
- Tell me about the last time you [insert problem], please.

From these basic questions, we can begin to understand the person's decision space. We can gain an understanding of how good the current solution is from the user's perspective and why this solution was chosen. Questions you could ask include the following:

- How well is your current way of solving [the problem] working for you?
- How are you doing with the solution?
- What is good about the solution from your point of view?

- What is less good about the solution from your point of view?
- How did you decide to solve [the problem] this way?
- Have you always done it this way?
- What have you done differently in the past?
- How did you decide not to do it that way anymore?

For more on formulating questions and using questioning techniques, refer to Chapter 6, Section 6.3.2.

These types of questions give you a good overview of the assessment of the current situation from the user's point of view, but above all, you also learn what has already been tried, what didn't work, and why it didn't work. This information helps you avoid stumbling blocks when designing new products.

Notice how, with these questions, we are asking about the situation at a fairly general level (low level of detail) and then jump from the present ("How are you doing right now?") to the past ("How was it before?").

We have not integrated the future yet, but it could be introduced. Since users have task and goal expertise but usually not design expertise (for a differentiation of expertise, see Chapter 6, Section 6.2.1), we advise keeping questions about the future rather general and not discussing them at a detailed/interface level. Of course, in some cases, users can also introduce design proposals. Whether this approach applies in your specific case should be assessed individually based on the target group.

For example, a possible question might be, "Imagine you had a magic wand and could change everything exactly the way you wanted. How would you solve [problem] in the future?"

The primary goal is not to obtain helpful hints for implementation but to understand the user's ideal image and then use this image to make independent decisions for development.

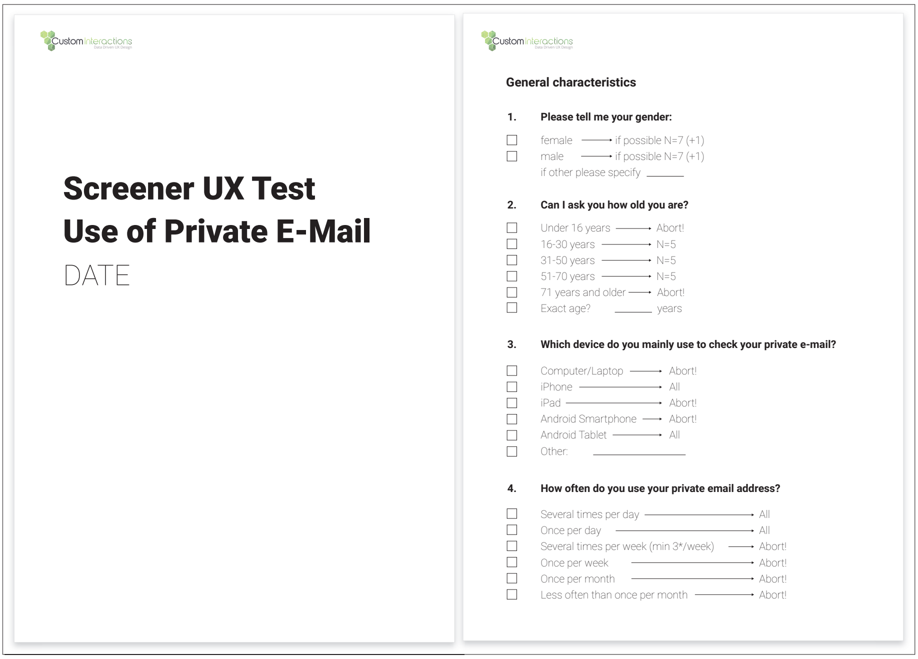
We then approach the individual aspects step by step and interview users about tasks and work steps until we reach the detailed level. The guide provides the basic framework for the interviews. Nevertheless, we recommend that you plan the interviews as semi-structured only so that you can add questions at any point during the interviews that are not in the guide but help capture the context of use, the goals, and the tasks.

The Preparation and Implementation

If you have never conducted an interview before, we recommend you practice conducting it a few times beforehand. Your goal should be to be completely neutral and interested when conducting it, just as a researcher would be. Remember: you want to gather knowledge, so you're asking questions. You do not want (at least not in this situation) to explain, tell, or be praised.

In addition, the rehearsal helps you to get a feel for the time required for the interview. This estimate is important so you can tell your potential participants how much time they will need to invest. You can conduct the practice interview with the actual target group. If, after the rehearsal, you feel that the interview was not well conducted and the results were affected, do not use it.

The right interview participants and the right questions are essential for your interviews to succeed. Even if you haven't defined your focus yet, you may already know who you would need to talk to. Consider "What makes a person who can help me learn what I want to learn?" How do you recognize a person with these qualities? We now create what we call a *screener*, which is a document that contains all the questions that determine whether or not we are talking to the right person, as shown in Figure 9.9.



Custom Interactions
Data Driven UX Design

Screener UX Test
Use of Private E-Mail
DATE

Custom Interactions
Data Driven UX Design

General characteristics

1. Please tell me your gender:

☐ female → if possible N=7 (+1)
☐ male → if possible N=7 (+1)
if other please specify ______

2. Can I ask you how old you are?

☐ Under 16 years → Abort!
☐ 16-30 years → N=5
☐ 31-50 years → N=5
☐ 51-70 years → N=5
☐ 71 years and older → Abort!
☐ Exact age? ______ years

3. Which device do you mainly use to check your private e-mail?

☐ Computer/Laptop → Abort!
☐ iPhone → All
☐ iPad → Abort!
☐ Android Smartphone → Abort!
☐ Android Tablet → All
☐ Other: ______

4. How often do you use your private email address?

☐ Several times per day → All
☐ Once per day → All
☐ Several times per week (min 3*/week) → Abort!
☐ Once per week → Abort!
☐ Once per month → Abort!
☐ Less often than once per month → Abort!

Figure 9.9 Example screener (excerpt).

We then ask the person these questions before the interview begins to determine if they are a potential interviewee. You don't necessarily have to create a screener, but you should be sure you're talking to the right people. If you interview the wrong people, you risk going down the wrong path entirely with your ideas and developments.

After you have determined who you want to talk to and what you want to ask, you can start right away. Now, conducting the interviews according to the guidelines and documenting all answers appropriately are all that remain. For information on conducting interviews, refer to Chapter 6, Section 6.3.3, where we've compiled various tips on the topic of *active listening*.

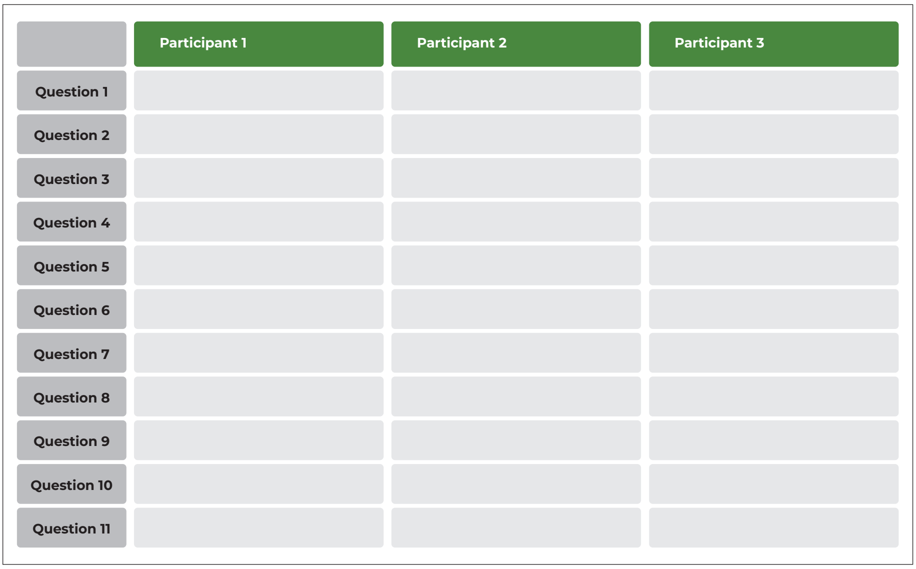
Appropriate documentation usually means recording interviews on video or on tape so we can evaluate them at leisure. If recording is not possible, we recommend conducting the interviews with two people: One person then conducts the interviews while the other focuses on documentation. If this scenario is also impossible, you should consider a system that allows you to take notes directly as much as possible. No matter what form of documentation you use, try to document or note what was actually said, not what you interpret to have been said. Interpretation can come later.

A word about implementation: Of course, you can also conduct most interviews remotely. If you're planning an interview exclusively without observation, an excellent interviewer can ask for approximately the same amount of information remotely as in person. Unfortunately, the two formats are not entirely comparable, as the delay or lack of view of the whole person often makes it more difficult to "tune in" to each other, and the conversation sometimes feels less natural than it would in a face-to-face meeting. You can counteract this to a large extent by conducting remote interviews exclusively with a camera: So, you see the person being interviewed, but they also see you.

Keep in mind, however, that the restriction to "people with webcam" already anticipates narrowing the target group to "more digital" people and that there may be differences to the core target group. This mistake in focus is then all the more serious if your system, product, or service addresses a "less digital" group of users.

The Evaluation

After conducting the interviews, we can analyze the data. We usually use a table to document all responses by entering one question per row and one participant per column, as shown in Figure 9.10.



	Participant 1	Participant 2	Participant 3
Question 1			
Question 2			
Question 3			
Question 4			
Question 5			
Question 6			
Question 7			
Question 8			
Question 9			
Question 10			
Question 11			

Figure 9.10 An evaluation table for interviews.

This table makes discovering similarities and differences in the answers easy. An even better approach is using colors and assigning a color to specific content. Then, you can see a quick overview of all the answers at a glance.

Whether you perform the evaluation digitally (e.g., with Microsoft Excel or another spreadsheet program) or on paper is entirely up to you. If you prefer analog, work with sticky notes and assign a color to each participant or topic. In most cases, we opt for digital documentation because it makes data easier to share and keep, which is especially useful when information needs to be added later in development, as often happens.

The results of the interviews give you information about the person's context of use: What are the person's current goals, tasks, and problems?

The preparation and documentation of the results can be done in many different ways. The easiest way is probably to summarize the answers given for each question. Then, for example, you would present the question and list all the answers below it, perhaps even with the number of mentions. At the same time, this way of presenting the results is often only of limited use since it contains the facts but leaves the interpretation to the reader. We recommend combining different questions into thematic blocks and documenting an interpretation of the results as a conclusion—in addition to the raw data (the original interview answers), which you hopefully keep anyway. This approach will speed up the use of the data significantly, and you won't have to go back to the data repeatedly.

For example, instead of simply listing the answers, you can write one or more summary sentences about a question. It could sound like this: "The respondents have in common that they place a particularly high value on freedom of choice in the products to be used. However, the selection criteria differ significantly between respondents. The most important criteria are price and availability."

You can make evaluation even easier by not just interpreting one question but by grouping several similar questions into a thematic block. Evaluating the interviews is a critical step in using the data. In addition to the evaluation process we presented earlier, you can also choose other evaluation methods such as *categorization*, *clustering*, or *content analysis*. To learn more about qualitative data analysis, check out the book *Qualitative Data Analysis: A Methods Sourcebook* (Miles et al. 2019), which deals with the topic extensively and in detail and is thus a good overview.

In many cases, interview data is then used to produce other kinds of information and to conduct further analyses, for instance, *task analysis* (Section 9.4.1), *user journey maps* (Section 9.4.2), and *personas* (Section 9.4.3).

Advantages and Disadvantages

The advantages of conducting interviews include flexibility and individuality. You can obtain a large amount of data on various topics within a relatively short time and at a low cost if you plan your interviews well. Open-ended questions can give you a lot of information and allow you to get a good picture of the context of use of your users.

Unfortunately, interviews have some downsides, for instance, the relatively large amount of time and effort required to gather, evaluate, and analyze the data. Unlike questionnaires or tracking methods (which we'll discuss later), evaluating open-ended questions takes a great deal of time. Depending on the level of detail of the evaluation, we usually calculate double to six times the duration for the interview evaluation. At the same time, a verbatim transcription (a word-for-word transcription) should probably be necessary for the fewest use cases outside of science.

Although qualitative interviews are not expensive, they still require a lot of time and effort. Thus, interview studies are often conducted with relatively few people, sometimes making it difficult to achieve good validity. Unfortunately, since the interview is a survey method, you can only gain information on aspects the users know. Unconscious aspects or implicit frameworks are often difficult or impossible to capture through interviews. Table 9.3 shows the profile for this method.

Profile: Interview/Jobs-to-Be-Done	
Brief description	A method of interacting directly with users and learning more about their roles, goals, needs, fears, desires, and attitudes by asking questions.
Advantages	■ Conducting interviews is very flexible and can be adapted to all questions. ■ Semi-structured and open-ended interviews provide opportunities to be very flexible in responding to the answers, thus maximizing learning. ■ Interviews are particularly well suited when there is still little information available on a topic, as open-ended questions allow topics to be raised by the interviewees themselves. ■ Interviews are relatively inexpensive to conduct.
Disadvantages	■ The effort required for a clean interview evaluation is relatively large. ■ Due to this effort, interviews are often only conducted with a few people, so the validity of the study suffers. ■ Only conscious aspects can be captured by a survey.

Table 9.3 Profile of the interview/jobs-to-be-done method.

9.3.4 Group Discussions/Focus Groups

Group discussions are the method of choice to obtain *"the broadest possible spectrum of opinions, views, and ideas about specific topics or products in a relatively short period"* (Berekoven et al. 2006, p. 96). In contrast to the interviews we described in the previous section, group discussions always occur with more people. Usually, between

6 and 12 people simultaneously participate in a focus group, as shown in Figure 9.11. A moderator or facilitator guides this group discussion.

Figure 9.11 A group discussion.

The Preparation and Implementation

As with individual interviews, a group discussion also follows a structure. We call this structure a *schedule*. Unlike an interview guide, the schedule contains not only the questions to be asked but also other details. The further details can, for example, refer to planned activities (e.g., group work, discussions, evaluations) but also concern the format of the activity ("Is it an activity that is carried out alone or in a group?") as well as the material required for it.

For us, successful focus groups include the following information in the schedule:

- Start time of activity
- End time of activity
- Precise questions and objective
- Type of activity (whole group activity, small group activity, or individual activity)
- Required material

Usually, a group discussion does not consist exclusively of asking and answering questions in a large group. Often, other activities are added in which the large group is divided into smaller groups or individuals who then complete smaller tasks (e.g., brainwriting) on their own. This mix of formats and tasks keeps the group lively and makes it easier to engage quiet participants.

The term *focus group* is frequently used, sometimes synonymously, sometimes with a slightly different focus (more activities instead of "only" discussions). Since very few group formats are exclusive discussion sessions and the term *focus group* is well established (and thus frequently requested), we'll always refer to it as a *focus group* in the remainder of this chapter. However, using these terms interchangeably is correct, even if the format is purely a group discussion.

Focus groups are always moderated. The skills needed to moderate a group differ significantly from those that characterize a good interviewer. The reason for this difference is that the most significant challenges in the focus group do not arise from the objectives but from the dynamics among the participants.

We recommend first learning to conduct individual interviews successfully before moving on to focus groups. On the one hand, you can gain confidence because you'll become familiar with comparable situations. On the other hand, conducting good interviews will also help you understand good moderation. Being able to conduct interviews is not a must, of course. For a focus group, moderation includes the following tasks:

- Clarifying objectives and creating an open atmosphere
- Making it clear that the discussion is not about consensus but about exchange
- Ensuring that all questions are answered
- Summarizing and visualizing answers
- Involving all participants (including the quiet ones)
- De-escalating and controlling "noisy" participants
- Managing time appropriately

Especially at the beginning of a focus group, the moderator has the crucial task of making all participants aware that the goal of the focus group is to learn about *different* views. Otherwise, getting participants to express opposing views can be difficult. Suppose the moderator succeeds in making this idea clear. In that case, a fruitful discussion will result, which, precisely because of the opposing opinions, will also reveal many different points of view and perhaps bring new topics to the surface.

The duration of a focus group is usually significantly longer than the duration of an individual interview—but the time left for each question or discussion with each participant is significantly reduced compared to interviews. A good duration for a focus group is 90 to 180 minutes. With longer sessions, concentration dwindles, and engaging in other activities (besides discussion) is almost mandatory to get good results.

Suppose you're working with people in business-to-business (B2B) environment. In that case, you'll probably notice that a more extended focus group is impossible or difficult simply because of the participant's availability, and thus, costs increase significantly.

In most cases, focus groups will occur in the afternoon or early evening if you focus on a working audience. Weekends are also an excellent time to simultaneously get several people in one place.

In terms of content, you can plan your focus group like a "normal" interview. However, due to the more significant number of people, you'll have to plan a longer time for each question so that fewer questions can be answered overall than in individual interviews. Again, we recommend clarifying the objective and learning needs beforehand and then creating a list of questions from rough to fine.

We recommend considering whether you get enough "depth" in your answers. Getting open and honest answers in a larger group can be difficult, especially regarding opinions and attitudes on private topics. However, feedback on problem situations or product ideas can be given without any problems in most cases.

The Evaluation

Documenting the results poses another challenge in focus groups. Unlike in individual interviews, it can be difficult to assign all the answers given to specific people without any doubt. It would help if you also thought about how to proceed with the documentation in advance. In most cases, creating videos in which all participants are easily recognizable is difficult or time-consuming, or you may have to record several videos. Therefore, we always conduct focus groups in pairs: One person has the task of moderating the focus group, and one person has the task of documenting the results.

For documentation purposes, moderation materials such as cards or sticky notes are usually used (for on-site focus groups). These can be labeled with small numbers before removing them to enable clear assignment afterward. Digitally, there are various approaches. One variant is to use tools where the authorship is visible on the contributions. This approach saves some moderation work but depends on all participants having access to the tool and being able to handle it well. Especially for "less digital" target groups, such tools can be a hurdle that leads to lower participation in the groups. To avoid this problem, we usually take care of the documentation for the participants.

The evaluation of the data follows the documentation. This evaluation is performed the same way as the evaluation of individual interviews but is usually faster because the quantity of information is much smaller. However, the goal is to convert the data obtained into other formats, such as *user journeys* (Section 9.4.2) or *personas* (Section 9.4.3).

Advantages and Disadvantages

One advantage of focus groups is that a broad spectrum of opinions and attitudes can be collected relatively quickly. Thus, this method is significantly less expensive than conducting individual interviews. In addition, a discussion with different people participating can bring to light attitudes that would not have become visible in an individual interview.

However, the disadvantages are just as obvious: By discussing in a large group, you get less information from each participant and can spend less time understanding individuals. Therefore, whenever we can, we use individual interviews and focus groups only when time or financial constraints make this unavoidable.

In addition, you're often also burdened with group effects when conducting group discussions. This effect comes into play especially if you have dominant people in the group who get significantly involved and thus shape the group discussion with their own opinions. To counteract this imbalance, you need strong moderation for group discussions.

A group discussion becomes more complicated when the topic is relatively private or where one's opinion may differ from the socially accepted one. Even then, group discussions produce significantly worse results than individual interviews, where there is more privacy and trust.

Table 9.4 shows the profile for this method.

Profile: Group Discussion/Focus Groups	
Brief description	*Group discussion* (or focus groups) is a collective term for methods in which several people are simultaneously interviewed about a specific topic. In addition to the actual questioning, the processing of smaller tasks in individual or group work can also be part of a focus group.
Advantages	■ Acquisition of many opinions and feedback within a short time. ■ Cost-effective compared to individual interviews. ■ Joint discussion can reveal opinions that would not have become apparent in individual interviews.
Disadvantages	■ Relatively short speaking time for each participant and thus less information in depth. ■ The danger of strong personalities dominating the group and thus shifting the group opinion in a particular direction. ■ Especially personal topics or socially less accepted opinions are less likely to be discussed honestly in a group discussion.

Table 9.4 Profile of the group discussion/focus groups method.

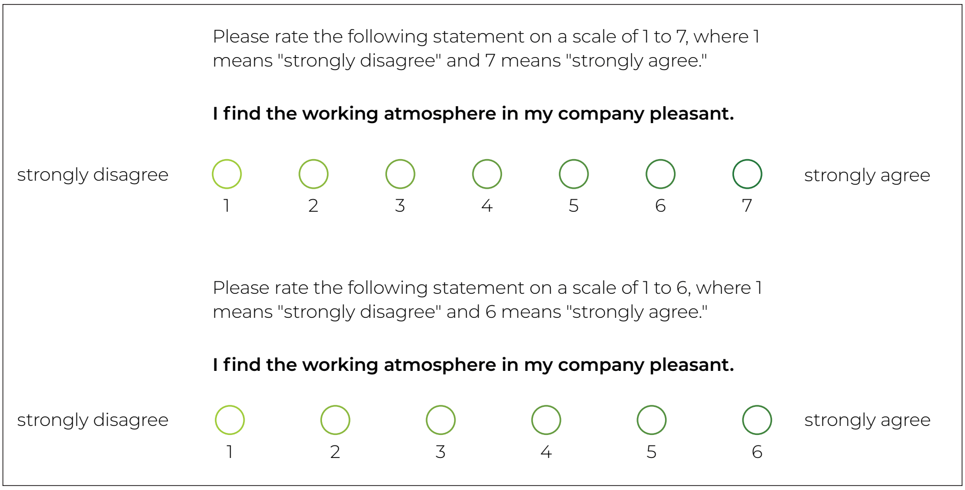
9.3.5 Quantitative Survey/Questionnaires

So far, to understand the context of use, we have focused exclusively on *qualitative methods*. Now, we'll introduce you to their quantitative counterparts, namely, the assessment of context of use based on *quantitative surveys*. In most cases, quantitative surveys mean questionnaires (online or offline) or conducting telephone interviews using structured questionnaires. To increase readability, we'll refer exclusively to *questionnaires* in this section.

Questionnaires are always helpful if you want to find out how often something happens. So, for example, if you have identified a problem in interviews, you could subsequently use questionnaires to determine for how many people this problem is relevant. Questionnaires are, therefore, often used in market research for market segmentation or as a barometer of sentiment, for example, before government elections.

For questionnaires to show their strengths, they are often designed in a completely standardized way. Questions or statements are put into a format that allows them to be answered on a scale. Common forms include the *Likert scale* (shown in Figure 9.12) or the *semantic differential* (shown in Figure 9.13).

With a Likert scale, the respondent indicates their agreement or disagreement with a statement in steps. Likert scales often have 5 to 7 levels, with typical scales ranging from 1 ("strongly disagree") to 5/7 ("strongly agree"). The more levels the scale contains, the more fine-grained agreement becomes possible. However, with more than 7 levels, uncertainty increases regarding the appropriate answer, and the additional information gained is relatively tiny.



Please rate the following statement on a scale of 1 to 7, where 1 means "strongly disagree" and 7 means "strongly agree."

I find the working atmosphere in my company pleasant.

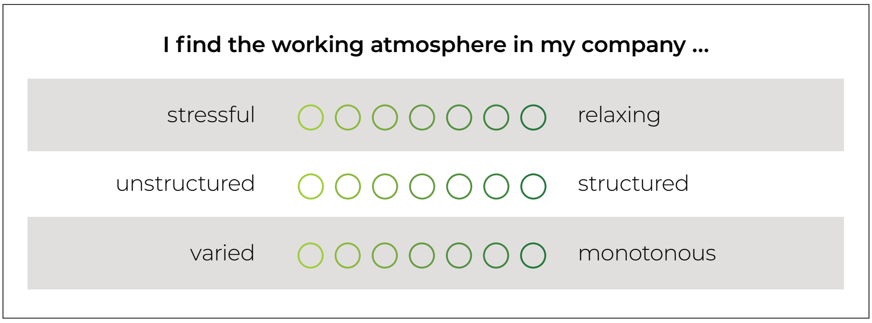
strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 strongly agree

Please rate the following statement on a scale of 1 to 6, where 1 means "strongly disagree" and 6 means "strongly agree."

I find the working atmosphere in my company pleasant.

strongly disagree ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 strongly agree

Figure 9.12 Likert scale with an even and an odd number of response options. An odd number of steps allows for a neutral answer in the middle. With an even one, a decision must be made.



I find the working atmosphere in my company ...

stressful	○ ○ ○ ○ ○ ○ ○	relaxing
unstructured	○ ○ ○ ○ ○ ○ ○	structured
varied	○ ○ ○ ○ ○ ○ ○	monotonous

Figure 9.13 Semantic differential.

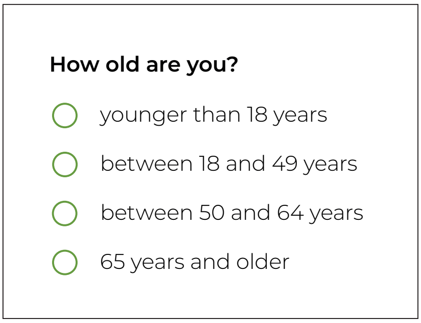
In most cases, Likert scales contain an odd number of alternatives and thus allow the responder to give a neutral answer. An example of a Likert question might be: “I find the working atmosphere in my company pleasant.”

A variant form of the Likert scale is the *semantic differential*. This survey method creates questions or statements that must be answered on a scale. Unlike Likert scales, however, this scale’s edges are not defined by values indicating (dis)agreement, but instead with adjectives representing opposite poles.

This type of question is particularly suitable when the “character” of the context of use, a task, or a person is to be captured. In later stages of development, semantic differentials could be used to assess a product’s character. Usually, semantic differentials combine a statement with several scales, for instance: “I find the working atmosphere in my company ... stressful/relaxing, unstructured/structured, varied/monotonous.”

In addition to Likert scales and semantic differentials, *multiple-choice questions* are also common in questionnaires, as shown in Figure 9.14.

In a multiple-choice question, a question is asked, and several answer options are given. Multiple-choice questions are probably best known from exams, which is why they sometimes have a bad reputation for being made only to confuse.



How old are you?

○ younger than 18 years
○ between 18 and 49 years
○ between 50 and 64 years
○ 65 years and older

Figure 9.14 Multiple-choice questions.

However, multiple-choice questions can be easy to quantify if different alternatives are known from interviews. Depending on the application, selecting only one answer option or several may be desirable. An example of a question where only one answer option is correct would be the following:

"How old are you?"

1. Younger than 18 years
2. Between 18 and 49 years
3. Between 50 and 64 years
4. 65 years and older

An example of a multiple-choice question where more than one answer can be correct would be:

"At what meals do you eat hot food?"

1. Breakfast
2. Lunch
3. Dinner

For the second question, the answer option "Other (namely):" would typically be added so that respondents can make additions. Multiple-choice questions are only helpful if you already know the possible answers well.

The questionnaire format limits you only a little in your choice of answer formats. Of course, you can also include *questions with a free response option* in your survey. However, use this option sparingly! Respondents often show a reduced willingness to provide much information as free text, which is associated with an increased effort.

In other words, most survey participants are lazy and don't want to write much. Who can blame them? You can avoid this trap by conducting surveys by telephone since you then do the writing for your participants. However, surveys conducted by phone also lose some of the core benefits, so you must weigh them individually for yourself and your project.

Advantages and Disadvantages

The use of standardized questionnaires has many advantages:

1. Questionnaires are inexpensive.
2. Standardized questionnaires are quick and easy to evaluate.
3. Questionnaires can be used regardless of location.
4. It is possible to interview many people with questionnaires.
5. Questionnaires enable a quantitative evaluation.

As you can see, the use of questionnaires can really pay off! On the other hand, some significant disadvantages do exist that you should consciously accept if you decide to use questionnaires.

For us, one of the most critical *disadvantages*—especially in the early phase of the context of use analysis—is the lack of additional information. You can only get answers to the questions you ask in the questionnaire. But especially in an early product development phase, when a lot of new knowledge is still being discovered, asking the right questions can be a real challenge.

In addition to asking the right questions, the *comprehensibility* and *different interpretations* of the questions in questionnaires are also essential. Evaluating both correctly is more difficult with a questionnaire format because answering the questions and analyzing the answers do not happen simultaneously. Thus, a misunderstood question could be noticed while analyzing the data—but not corrected, and therefore, the answer is lost. In a worse case, a misunderstood question might not be detected (for example, if the answer format is a scale), and incorrect data is retained.

In the case of questions with an “other” option, you can at least still detect misunderstandings. To avoid this trap, when developing psychological questionnaires that are to be used regularly and not once for product development, many attempts are made for the correct wording, and the same questions are also asked in different ways to perform a plausibility check between the answers. You’ll have to decide whether the effort is worth it for your project.

Another disadvantage of using questionnaires is *dishonesty in response*. Often, participation in questionnaires is mapped via panels. *Panels* are groups of people who have agreed to participate in surveys. Participants receive payment for surveys, so there is a relatively high interest in participating in as many surveys as possible. Depending on how well the individual panels are maintained and administered, participants may simply “click” answers together quickly. Filtering out these incorrect responses from the volume of data in a questionnaire is a challenge. To avoid these problems, several approaches include *plausibility checks* (comparison of answers to similar questions) or searching for *answer patterns* (e.g., “always checked 1” or “always checked 5”). However, this check also significantly increases the overall effort in the evaluation. Alternatively, well-maintained panels can be used, which, however, is usually associated with higher costs for implementation.

Tip: Dealing with Online Panels

If you’re looking for a suitable online panel, you’ll find one for almost every topic. When selecting panels, we recommend you pay attention to the panel’s description on the respective website. The more precisely a panel is geared (e.g., to a particular profession), the more likely you’ll find the right target group for your survey.

We recommend that you generally use panels that allow participants to be screened out (i.e., by ending the survey if inappropriate answers are given at the beginning) in conjunction with a one-time link for each participant. In this way, you can prevent people from trying different answers multiple times until they can successfully pass the screener and get paid for participation.

Well-maintained panels periodically prompt participants to update their data and allow companies using the panels to report inappropriate or questionable responses via an anonymous participant ID. This approach keeps the database "fresh" and free of unmotivated individuals.

The last—slight—disadvantage is the *amount of information* that can be obtained with questionnaires. In most cases, the willingness to answer a questionnaire is relatively low. Willingness also affects the patience people have when filling out questionnaires. We recommend keeping your questionnaires as short and as concise as possible and allowing no more than 5 to 10 minutes of survey time.

This restriction alone limits the scope of possible questions. Alternatively, you can try to accommodate more questions and make only a few mandatory for online questionnaires. In this way, the overall questionnaire can be longer without necessarily increasing the effort for each participant. Of course, this will cause you to lose answers in cases where participants only answer mandatory questions.

From our presentation so far, we hope you take away that we view questionnaires as critical for a context of use analysis in most cases. Questionnaires undoubtedly have a high value for every company in later product phases—for example, to obtain an actual status after the launch and to track a possible change in customer satisfaction. During development, however, questionnaires are usually advantageous only at a few moments and even then, never as a standalone method.

One of the few moments when questionnaires are extremely useful in the early stages of development is when developing *personas*. But for this task, too, we usually take a two-stage approach, starting with interviews and observations and then using questionnaires to test our findings quantitatively. Of course, we're not saying questionnaires can't be used advantageously; we just don't feel we've done it well yet.

Table 9.5 shows the profile for this method.

Profile: Quantitative Survey/Questionnaires	
Brief description	Conducting a written survey using questionnaires (on paper or digital) to obtain information about the prevalence and frequency of certain attitudes. A quantitative survey almost always occurs as a structured survey, in which both the questions and the answer options are predefined.

Table 9.5 Profile of the quantitative survey/questionnaires method.

Profile: Quantitative Survey/Questionnaires	
Advantages	■ A questionnaire enables the questioning of many people in a short time. ■ A questionnaire enables the quick and easy evaluation of data. ■ When conducted digitally, a questionnaire can be used quickly worldwide. ■ Implementing questionnaires is usually highly cost effective. ■ A questionnaire allows statements about the frequency or spread of opinions due to a large number of participants with a representative group of participants.
Disadvantages	■ You'll only receive answers to the questions you ask, which is unsuitable if you have little prior knowledge of a topic. ■ Due to the lack of the opportunity to ask questions, misunderstandings can occur. ■ Detecting and filtering dishonest answers from the mass of responses is challenging. ■ Participants often show little willingness to answer open-ended questions in writing. ■ Only short written surveys (recommended maximum duration of 10 minutes) are possible without losing response quality.

Table 9.5 Profile of the quantitative survey/questionnaires method. (Cont.)

9.3.6 Contextual Inquiry

Contextual inquiry is one of our favorite methods for a context of use analysis. Although the method's name sounds complicated, this method is nothing more than a combination of field observation and interview. To complicate things, the method can also be called *field observation with a downstream interview*, while the term *contextual interview* is sometimes used (but rarely in the usability and UX area). To keep things simple, we'll stick to *contextual inquiry*.

We want to point out that we do not mean participant observation with contextual inquiry! So, this activity is not about you performing the tasks and being directly involved in the processes. Contextual inquiry can be useful in a business-to-consumer (B2C) environment with relatively simple applications. For most B2B use cases, however, trying to experience the system, product, or service yourself is not useful. You should create a separate bonus step if you want to do that as well.

This gap exists because it is challenging for outsiders to "feel their way" into the web of tasks, goals, and organizational framework conditions partly responsible for a person's actual activity. Attempting to do so even carries the risk that, after performing an activity once, you'll see yourself as a kind of expert in the future—with a completely different level of knowledge and training than the actual users. However, we recommend focusing first on understanding your counterparts with their goals and tasks and building upon that foundation.

The Implementation

In a contextual inquiry, you go on-site like in a field observation. Unlike in the latter, however, you're not just a passive observer but instead actively question the person you observe, as shown in Figure 9.15.

Figure 9.15 A typical contextual inquiry situation.

You can interview in parallel with the observation or as a follow-up. Of course, the most efficient way is to do both tasks in parallel. In this context, however, you may quickly find that you exert a relatively significant influence on the interviewee and their behaviors through your questions and intervention. Thus, if neither you nor the interviewee is under time pressure, it may make sense to separate the two activities and be a passive observer before jumping straight into an interview.

You can apply the same rules for preparing the contextual inquiries as for preparing the field observations and interviews. However, since both parts influence each other, you should be prepared to create a list of questions for the interview part and add many live questions that emerge from the observation situation. This opportunity to formulate questions on the fly is precisely one of the strengths of contextual inquiry. If done correctly, this approach can save you an iteration loop where you must painstakingly bring together the information from observation and interview. This method is an absolute insider tip, especially at the beginning of a project, to gather a large amount of information as cheaply and quickly as possible.

As you accompany the observation with questions, the order of questioning will change compared to a pure interview. Our recommendation for conducting "pure" interview studies is to start in the present with general parts, then move to how these parts were solved in the past and (if applicable) how they should be in the future before asking more detailed follow-up questions afterward. In a contextual inquiry, since you're directly observing actions and asking follow-up questions about those actions, in most cases, you'll start in much more detail and stay only in the present. Typical questions include the following:

- What did you just do?
- What kind of information did you just look up?
- What did you ask your colleague about?
- Why do you need the following information?

So, the questions are essentially questions that arise during the observation that make the person's current action understandable for you. Afterward, you do move on to general questions. If you try to integrate the general and broad questions into the observation, you'll likely end up with the following results:

1. The implementation takes much longer, as much time is spent on discussion and reflection.
2. Execution changes, if necessary, as one's actions become more reflective by questioning the broader context.
3. You'll be able to observe fewer actions because the total time is not likely to or can become longer, but more time will be spent on questions.

Thus, to use this method's full potential, our clear recommendation is to focus on comprehension questions about task execution during the observation and to capture the overall context only afterward.

The Evaluation

As a result, you usually have documentation of the observation (either as a transcript in a table format, as video documentation, or a combination of both) and a recording of the interviews (either digital or analog). To evaluate the results, you can then apply all the methods that you would otherwise use to evaluate pure interviews or field observations.

In the simplest form, a contextual inquiry only gathers an enumeration of actions and their frequencies or durations (whereby durations may be only conditionally correct due to the interview) and the individual answers to the questions. However, it is often advisable to further summarize the data obtained and make it more accessible by transferring it to another type of presentation. Methods such as *task analysis* (Section 9.4.1) or the *user journey* (Section 9.4.2) can help in this process.

Advantages and Disadvantages

The advantages of a contextual inquiry can be seen in the reduced time required for its implementation and the increased information gained compared to pure interviews and field observations.

In contrast to conducting a pure field observation, which only reveals the *what* (i.e., the visible action) or conducting pure interviews, which only reveal the *why* (i.e., attitudes, goals, and reasons), contextual inquiry allows you to explore both the what and the why in one step. Not only can you save time, but also your participants. By asking questions during the observation, you gain a much deeper insight and understanding than would be the case with a pure observation or a pure interview.

On the other hand, you run the risk of influencing their behavior through one's participation, and thus influencing the accuracy of the data gathered. Furthermore, this method requires a relatively high evaluation effort due to its qualitative character and the high volume of data generated by the combination of observation and interview. In addition, gaining access to participants can be quite difficult, especially if the topics are in the private sphere or in a B2B company.

Despite these disadvantages, we strongly advise using this method, especially at the beginning of a project. We are unaware of any other method that provides such a comprehensive and in-depth understanding of the context of use in a comparably short time. Table 9.6 shows its profile.

Profile: Contextual Inquiry	
Brief description	Contextual inquiry is a mixture of field observation in the actual context of use and simultaneous or downstream interviewing of the observed persons.
Advantages	■ Capture both objective and subjective information in one pass. ■ Simplified acquisition of participants compared to two separate appointments. ■ Better understanding compared to using only one method.
Disadvantages	■ High risk of influencing the respondents' behavior through the interview. ■ Relatively high evaluation effort since both observation and interview data are generated. ■ Like with field observations, access issues may exist, especially in private or B2B environments.

Table 9.6 Profile of the contextual inquiry method.

9.3.7 Analysis of Quantitative Usage Data

All the methods presented so far rely on you generating data for the context of use analysis that did not exist before. In this section, we'll introduce you to some methods that rely on you analyzing existing data and using it for the context of use analysis.

The first "method"—namely, the analysis of quantitative usage data—represents not so much a fixed procedure as a super-category of data to be collected and analyzed. Most companies now have a large amount of data on the use of their products. This wealth of data is especially the case if these products operate on the cloud or as a web application and reside in some form on the vendor's servers. Most companies we talk to know the treasure trove of data available here. Yet we very rarely encounter companies that actually analyzing this usage data.

By *usage data*, we mean all the data that can also be used as product-related key performance indicators (KPIs) for calculating return on investment (ROI), as described in Chapter 3, Section 3.2. These KPIs could, for example, include the following measurements:

- Frequency of use of individual features
- Scope of use of individual features
- Usage time of individual features

Analyzing usage data on its own is difficult in most cases. We ourselves struggle with the question of whether a feature is not being used because it is not needed or because it is not found or not understood. Especially in the context of use analysis, an essential goal is to understand what is happening and why, and thus, we recommend *trying to create a big picture whenever you can.*

You should use all the quantitative usage data you can access to formulate your initial assumptions. This step can be done either at the beginning of the project in the stakeholder workshop using already existing data or afterward to start a first check of still unproven assumptions. Afterward, use qualitative methods to check whether you're right and to discover new aspects.

For example, suppose your support team keeps getting requests for a particular feature. Still, that feature is already implemented, and the usage data shows that the feature is rarely used. In this case, you can assume with relative safety that this issue is a *usability issue*.

The reason for this usability issue may be that users cannot find a function, expect it under a different name, or are unsatisfied with the scope of a function and its implementation. With a probability of 95% (our own unscientific estimate), you'll encounter a usability problem in some form.

At this point, most people involved in development who have a technical background tend to tweak the feature itself. Further options are often added to implement the feature "correctly." However, tweaking the feature could be the completely wrong approach if—as is so often the case—users find something too complex or inappropriate for their task. Therefore, again our recommendation: Determine *why* something is desired and use a qualitative method for this determination, for instance, an *interview* (Section 9.3.3), a *contextual inquiry* (Section 9.3.6), or a usability and UX test (see Chapter 12, Section 12.7).

The Implementation

A typical approach to analyzing quantitative usage data is to *look for patterns in the usage data*. You can search for patterns in usage (e.g., "Users who use function A also always use function B"). You can also search for specific patterns, subdividing them by usage groups. *Subdividing into user groups* is always helpful if you already have assumptions about correlations and want to check them against the usage data.

An example of a typical assumption is that large companies use products differently than small companies or that older people use products differently than younger people. Of course, which assumptions make sense in your field depends heavily on your system, product, or service and the market.

Obviously, your data analytics capabilities are directly related to the *availability of the data*. For example, when selling to businesses, in only rare cases will you actually have data on the age of the end customers. In most cases, analyzing quantitative usage data is only possible if you plan another device generation or an evolution of an existing product. If you're setting up a completely new product or entering a new application area, you'll have little or no usage data to fall back on.

Advantages and Disadvantages

The advantages of analyzing existing usage data are pretty straightforward: Usage data reflects actual behavior and is, therefore, very valid. In addition, the volume of usage data is usually relatively high, so you can analyze many different kinds of data and test a wide range of assumptions. In addition, the data—especially if technically prepared—is already readily available, so no additional effort is required to generate it, either by you or your users. Moreover, the data is derived from usage so that it can be viewed repeatedly at different times.

On the negative side, however, analyzing usage data is only possible if you already have a product on the market that you want to optimize. For the context of use analysis phase, a critical consideration to note is that pure usage data can usually only explain part of it (the what), and much of the information is not understandable in purely quantitative terms. In turn, the lack of understanding means that you'll have problems achieving the desired usability and UX when designing your system, product, or service—simply because you do not understand sufficiently well the tasks, goals, and framework conditions under which work is being done.

Nevertheless, do not miss any opportunity to look at existing usage data. However, to get the most out of the data, combine quantitative usage data with other data, for instance, from ticket analysis (Section 9.3.9), web tracking (Section 9.3.8), and qualitative interviews (Section 9.3.3). Another obvious disadvantage is that you can only evaluate usage data if you have already collected it during use. Thus, making technical changes to the product may be necessary, which can mean a considerable amount of work afterward.

Table 9.7 shows the profile for this method.

Profile: Analysis of Quantitative Usage Data	
Brief description	Using existing data from the use of a product to gain new insights into the use of solutions without having to collect new data.
Advantages	▪ Data reflects actual usage behavior. ▪ No effort is required for additional data collection—neither by you nor by users. ▪ Large amounts of data are available for evaluation; thus, often, many hypotheses can be tested. ▪ Continuous data collection makes it possible to repeat the evaluation at different points in time.
Disadvantages	▪ This method only works if there is already a solution/predecessor solution. ▪ Poor understanding of the why behind usage data—resulting in little evidence of how to design the system, product, or service better. ▪ Requires the technical prerequisite that data can be collected.

Table 9.7 Profile of the analysis of quantitative usage data method.

9.3.8 Web Tracking/Click Analysis

The Internet makes your job as a usability and UX designer much easier today than it would have been a few years or decades ago. Using tracking pixels and cookies, you can track the movements and actions of web users step by step, as shown in Figure 9.16.

If you own a website—or want to optimize its use—web tracking gives you extensive information about the actual behavior of your users.

Typical information that you can gather from tracking websites includes:

- The number of websites visitors per day, week, month, and year
- Entry points (on which page do users start?)
- Dwell times (how long are users on a particular page?)
- Click paths (how do users move through your website?)
- Conversion rate (what percentage of users perform a particular action?)

- Drop-off points (where do users leave your website?)
- Areas of interest (what information is given special attention?)

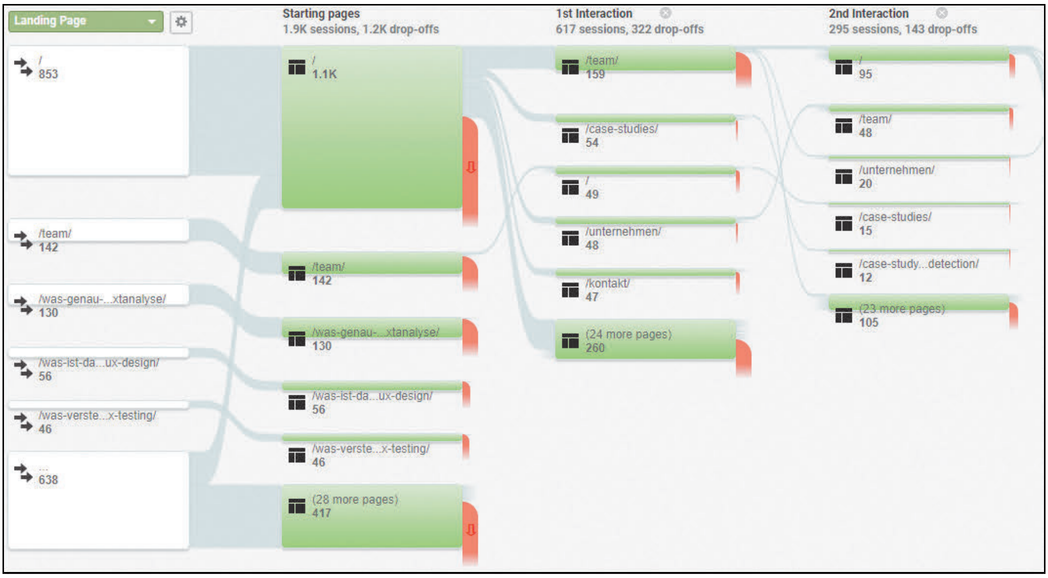

Landing Page
Starting pages
1.9K sessions, 1.2K drop-offs
1st Interaction
617 sessions, 322 drop-offs
2nd Interaction
295 sessions, 143 drop-offs
/
853
/team/
142
/was-genau-...xtanalyse/
130
/was-ist-da...ux-design/
56
/was-verste...x-testing/
46
...
638
/
1.1K
/team/
142
/was-genau-...xtanalyse/
130
/was-ist-da...ux-design/
56
/was-verste...x-testing/
46
(28 more pages)
417
/team/
159
/case-studies/
54
/
49
/unternehmen/
48
/kontakt/
47
(24 more pages)
260
/
95
/team/
48
/unternehmen/
20
/case-studies/
15
/case-study...detection/
12
(23 more pages)
105

Figure 9.16 Website analysis in Google Analytics where you can track exactly how users have moved around your website.

If you have clearly defined paths that people are supposed to follow on your website (e.g., in the case of an online store), you can quickly determine at which point your *funnel* breaks. This term from sales describes the process of turning unknowns into customers. The idea behind the sales funnel is that people follow a defined path until they end up taking the closing action (e.g., making a purchase, signing up for a newsletter, making a phone call, downloading a file). The funnel's shape represents how the number of people who work their way forward step by step usually becomes smaller and smaller, as shown in Figure 9.17.

When analyzing web data, you usually compare your ideal click path with the click paths actually taken by users. Thus, you can check at which point users "take a wrong turn" or end the interaction even though the action you were hoping for has not yet been taken. Once you have found the point at which significantly fewer people take the next step, you can begin with analysis and optimization.

We recommend you start analyzing the problems as far up the funnel as possible to use your resources as efficiently as possible. This approach also has the advantage of having more data in the subsequent funnel steps. For example, if you have 10,000 users in the first step, but only 2 of them arrive in the penultimate step, your database is weak at that almost final step—and optimization would not bring much in absolute terms.

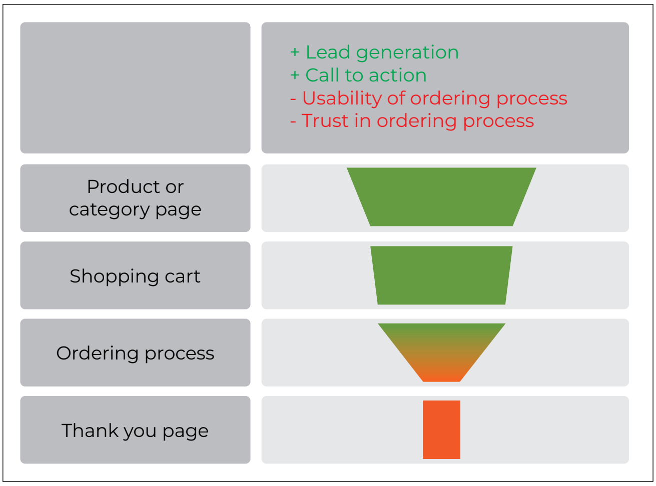


Figure 9.17 Example funnel for a web store. Depending on the point at which people drop out, problems can be analyzed and remedied in a targeted manner.

A More Detailed Example

Imagine you're the owner of an online Christmas decoration store, and you want to improve your sales.

As a first step, you could check how many people come to your website and from where. If you have advertisements (e.g., on Google), your funnel starts even one step earlier. Then, the best next step would be to examine which search terms your ads were shown on, who responded to them, how many people saw them, and how many clicked on them.

Once on the website, the next step is to check how long people stayed, what information they looked at, and where you lost many people for the first time. For example, if your ad was shown to 10,000 people, but only 5 people landed on your website, you should urgently analyze and optimize the quality of your Google ad first. Perhaps you discover that your ad does not match the expectation of your users. But if you could understand the problem at this point and then solve it later in the data-driven UX design process, you would significantly increase the probability of successful sales.

Similar to the analysis of usage data, the problem arises again that, although you notice something is not working, it is challenging to understand why people are not clicking based purely on click-through rates and paths.

Usually, in the web world, optimization is then performed using *A/B testing* (more on this topic in Chapter 12, Section 12.10). However, depending on your confidence in interpreting the data, we recommend combining data from website usage with qualitative data from interviews.

When a website is already in live use, you could also work with initial usability and UX testing (see Chapter 12, Section 12.7) for generating qualitative data instead of just interviews. This activity would give you direct insight into the stumbling blocks in using your store and the drop-off points in your website. This test is probably the best-known method for testing usability and UX and is a mixture of observation and interview. However, since usability and UX tests—especially for new developments—can often only be used in the “final” phase, we’ll present them again in detail there.

In general, we recommend using a mix of usability and UX testing and interviewing to gain the best possible understanding of the context of use and prevailing issues.

Advantages and Disadvantages

Since web tracking or click path analysis is a subset of “usage data analysis,” the advantages and disadvantages largely coincide. In addition to the information described in Section 9.3.7, we would like to add the following: On the positive side, an extensive database reflects users’ actual behavior. On the other hand, web tracking or the analysis of click paths on web pages is only feasible if a web page or web service is already provided. Also, the purely quantitative data this mainly generates does not provide any information about the background for a behavior.

Table 9.8 shows the profile for this method.

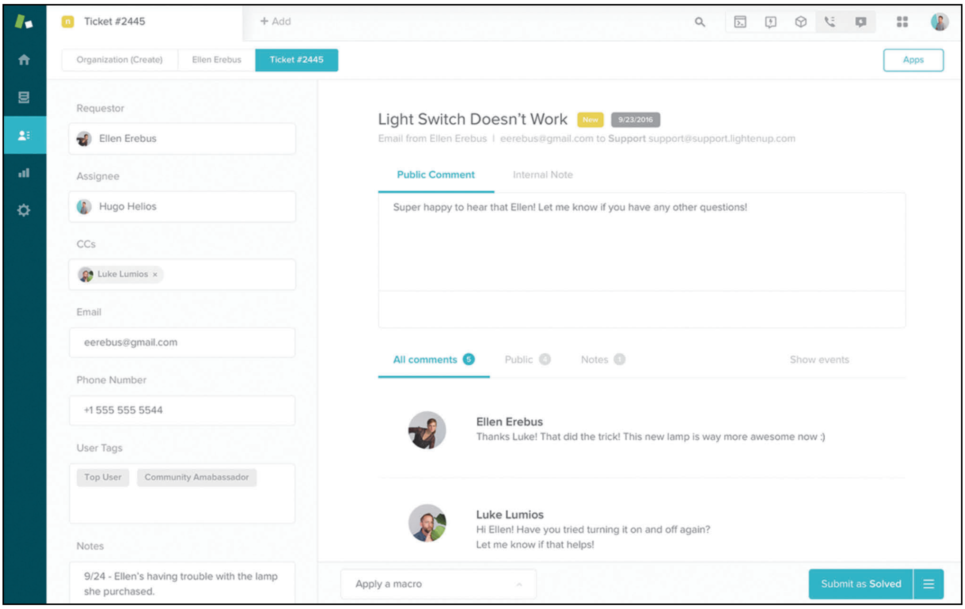
Profile: Web Tracking/Click Analysis	
Brief description	Tracking of users’ movements and actions on websites and in applications.
Advantages	▪ The data reflects actual usage behavior. ▪ No effort is required for additional data collection—neither by you nor by the users. ▪ Large amounts of data are available for evaluation, so many hypotheses can often be tested. ▪ Continuous data collection makes it possible to repeat the evaluation at different points in time.
Disadvantages	▪ This method only works if there is already a web application. ▪ Poor understanding of the why behind usage data—resulting in little evidence of how to design the system, product, or service better. ▪ Requires the technical prerequisite that data can be collected.

Table 9.8 Profile of the web tracking/click analysis method.

9.3.9 Ticket and Feedback Analysis

The *ticket analysis* method is not a formal method that you’ll find in various usability and UX books, but this method is a logical step if you want to analyze your usage context in a data-driven way. In general, the core of the methodology presented in our context is the systematic analysis of existing customer feedback in a company. The two special cases we'll explicitly address are the *analysis of support tickets* and the *analysis of customer feedback provided in writing*. However, you can also apply the same idea if you have other or additional types of customer feedback available (e.g., call center logs).

Let's start with support tickets to clarify the idea. In most cases, you have a support tickets system if you have an IT product on the market where users can report problems, as shown in Figure 9.18.



Requestor: Ellen Erebus

Assignee: Hugo Helios

CCs: Luke Lumios ×

Email: eerebus@gmail.com

Phone Number: +1 555 555 5544

User Tags: Top User, Community Amabassador

Notes: 9/24 - Ellen's having trouble with the lamp she purchased.

Light Switch Doesn't Work

Email from Ellen Erebus | eerebus@gmail.com to **Support** support@support.lightenup.com

Super happy to hear that Ellen! Let me know if you have any other questions!

Ellen Erebus
Thanks Luke! That did the trick! This new lamp is way more awesome now :)

Luke Lumios
Hi Ellen! Have you tried turning it on and off again?
Let me know if that helps!

Figure 9.18 The Zendesk ticket system.

Often, the submitted tickets are feature requests (i.e., a request for missing functionality or usage issues and bugs). Most of the time, tickets are automated or manually grouped and assigned to support staff. In most companies, ticket processing is well monitored. This data includes the number of incoming tickets, the processing speed, and satisfaction ratings for the processing.

What we rarely see, however, is an analysis of the *content of the tickets*. Although the tickets are grouped and then assigned to specific people for processing, this person then "only" subsequently selects a suitable solution path based on an existing knowledge database or develops their solution. We are practically only aware of cases focusing on the problem and its solution. This focus is understandable; after all, solving a specific problem is the primary task of a support employee.

However, this focus also wastes potential. After all, tickets contain information that can help better understand the context of use if tickets are systematically and appropriately recorded.

Therefore, if the time and opportunity are available, then you should periodically view a portion of the support tickets and ask the following questions:

- For which task did the user want to use the system, product, or service?
- What is the goal behind its use?
- What general conditions of use become clear from the ticket?
- What functions do users want, and for what reason?
- Are the user's tasks, goals, and frameworks included in your system, product, or service, or does the user move outside the framework you have in mind?

Attention: Distortions!

You can learn a lot about the context of use from ticket data. However, beware of one catch: Usually, no formal requirement exists that the context of use information be submitted with a ticket. Therefore, bias in the data may exist. For example, one user group could be particularly willing to provide information while another group forgoes such information altogether. Uncovering this type of bias using only ticket data is difficult. But if this bias is not discovered, false assumptions can easily arise.

Of course, you can use ticket analysis not only to understand the context of use for an existing system but also to analyze tickets regarding submitted issues. However, if you do that and analyze more at the task and surface level, you're more in the realm of testing an application at the end of the cycle.

As with all other methods that use actual data from users, information that can be used for context of use analysis and information that can be used for testing the product itself merge seamlessly. And you face the challenge that analyzing ticket data is only possible if you already have a live ticket system.

Another type of data that is often present in existing systems, products, and services is *user feedback data*. This data can be collected through dedicated feedback channels, such as a form on the website, but it can just as easily be feedback emailed to a contact address or shared on forums or social media. In feedback analysis, the questions you should try to answer to get input for the context of use analysis are again precisely the same as in ticket analysis, namely:

- For which task did the user want to use the system, product, or service?
- What is the goal behind its use?
- What general conditions of use become clear from the ticket?
- What functions do users want, and for what reason?
- Are the user's tasks, goals, and frameworks included in your system, product, or service, or does the user move outside the framework you have in mind?

Feedback analysis has a unique feature compared to support tickets: Depending on the product, the feedback may be publicly available, at least in part. For example, you can easily search social media for your system, product, or service—but you can also do the same for *competitors*.

This approach gives you a disproportionately more significant amount of feedback to analyze than would be the case if you were looking only at your own solutions. The important thing in this context is that you shouldn't select the competitors you look at by industry, but by whether they solve the same problem for their customers and help implement the same goals. In this way, if you use public data, feedback analysis is a method of "analyzing existing usage data" that you can use even if you do not yet have a product.

Depending on your application field, some institutions actively collect feedback for you. For example, as a manufacturer in the medical device field, you can access the *MAUDE database*, which provides interested parties with data on submitted incidents related to medical devices in the US. This type of feedback is, of course, unique in that it relates exclusively to negative feedback. Still, nevertheless, the database provides a large amount of feedback that would not otherwise be accessible. The MAUDE database can be found at *https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfmaude/textsearch.cfm*.

Advantages and Disadvantages

In summary, the advantages of ticket analysis are the large amounts of data and the possibility of inferring, at least indirectly, further information about the context of use. The fact that the support tickets and feedback forms are filled out during regular use means that they also reflect the actual problems and wishes of the users and do not result from an artificial study situation. In this way, you can also prioritize issues based on the frequency of specific topics alone.

Because the data is generated continuously, you can also repeat the evaluation again and again and see whether the user's needs and problems change over time. With good data maintenance, this method even allows you to conclude the effect of product adjustments that have been made. Another advantage is that feedback data on many products is often publicly available. Therefore, you can not only analyze your data but also draw on the data of market companions.

On the negative side, ticket or feedback analysis can lead to a bias in the feedback since it is primarily negative feedback that you'll receive since dissatisfied people are the users that primarily seek out feedback opportunities.

As with all other methods that use existing data for the analysis, you also need an existing product with a functioning ticket system already. In addition, the analysis of existing data can become relatively time consuming if the presorting of tickets is primarily technical and does not support the identification of usage problems and sorting according to usage.

Table 9.9 shows the profile for this method.

Profile: Ticket and Feedback Analysis	
Brief description	Evaluation of existing support tickets and customer feedback to identify and analyze problems, requests, and requirements for the system, product, or service.
Advantages	■ The data reflects the actual problems and wishes of the users. ■ No additional data collection effort is required beyond filling out support requests or feedback forms. ■ Large amounts of data are available for evaluation so that critical areas can be easily identified. ■ Continuous data collection makes it possible to repeat the evaluation at different points in time. ■ You can even access competitor product data via support forums or public feedback channels.
Disadvantages	■ Feedback will be biased towards aspects of the product that are perceived as negative, as these negative experiences predominantly end up in support. ■ Only works if there is already a solution/predecessor solution. ■ Elaborate filtering of tickets to identify which ones are even usage related.

Table 9.9 Profile of the ticket and feedback analysis method.

9.4 Analytical Methods of the Context of Use Analysis

The presented methods are designed to help you understand the context of use and collect data to help you develop new assumptions and refute or confirm existing ones. However, to further analyze this data and make it accessible to third parties not involved in collecting it, we would like to introduce you to three more methods to help you do this.

9.4.1 Task Analysis

Task analysis is a method that breaks down the entire user task into individual substeps and then considers these substeps individually (see also IEC/TR 62366-2:2016). *Task analysis* is a method not used to collect data about the context of use but to analyze the collected data.

Task analysis often occurs as a follow-up to a field observation (Section 9.3.2) or a contextual inquiry (Section 9.3.6). So, whenever you have previously collected information about the task and its execution, you can initiate a task analysis.

The basic idea of task analysis is straightforward: You take an overall task and break it down into its subtasks. For example, imagine you would like to re-analyze the activity "make coffee," as shown in Figure 9.19.

Then, your task analysis might reveal the following:

- **Main task:** Make coffee
- **Subtasks:**
  - Switch on the coffee maker
  - Get coffee cup
  - Place coffee cup under coffee nozzle
  - Wait for heating up
  - Decide on portion size
  - Select portion size
  - Wait for the coffee to be prepared
  - Take the full coffee cup

Figure 9.19 "Making coffee" as an example of a task analysis.

From this simple example, notice how even a simple and short task can be divided into many parts. Of course, we could also list more steps if the coffee is further processed at the end and, for example, milk foam is produced in parallel. Additionally, in this example, we have entirely left out the typical "disasters" of a coffee machine: We don't need to rinse the machine, the beans are refilled, there is always enough water, and the container for collecting the coffee grounds is emptied often enough.

Nevertheless, we come to 8 steps in this example alone. If you look closely at these steps, not all of them are actions in the observable sense. While you can observe how the coffee machine is switched on, you cannot see a portion size being decided upon. You only get this step indirectly: by looking at the next step (choosing the portion size) and inferring what must have happened before (in which case you're making an assumption) or by asking the users what happened here and why. No matter how you go about it, a good task analysis includes all of the user's tasks, whether in the external world or in the internal world.

When you use task analysis in the context of use analysis, you usually represent an as-is state. In other words, you describe the process you have seen for completing the task. Since, in most cases, you have not only observed one person, you'll usually see different processes. To use the task analysis for the next steps, it becomes even more important to be able to compare the individual processes. Depending on what is more straightforward, you could use a list format, for instance, in Microsoft Excel, or a more graphical representation like a Unified Modeling Language (UML) flowchart, as shown in Figure 9.20.

Both forms of presentation will work quite well for your own viewing. If you want to discuss the process with others, we recommend the graphical representation of the content because it is more accessible to many people. For us, a simplified notation has worked well, where we only distinguish between boxes (actions), diamonds (decision points), and lines (connecting paths between steps). In most cases, these simple flowcharts are perfectly adequate.

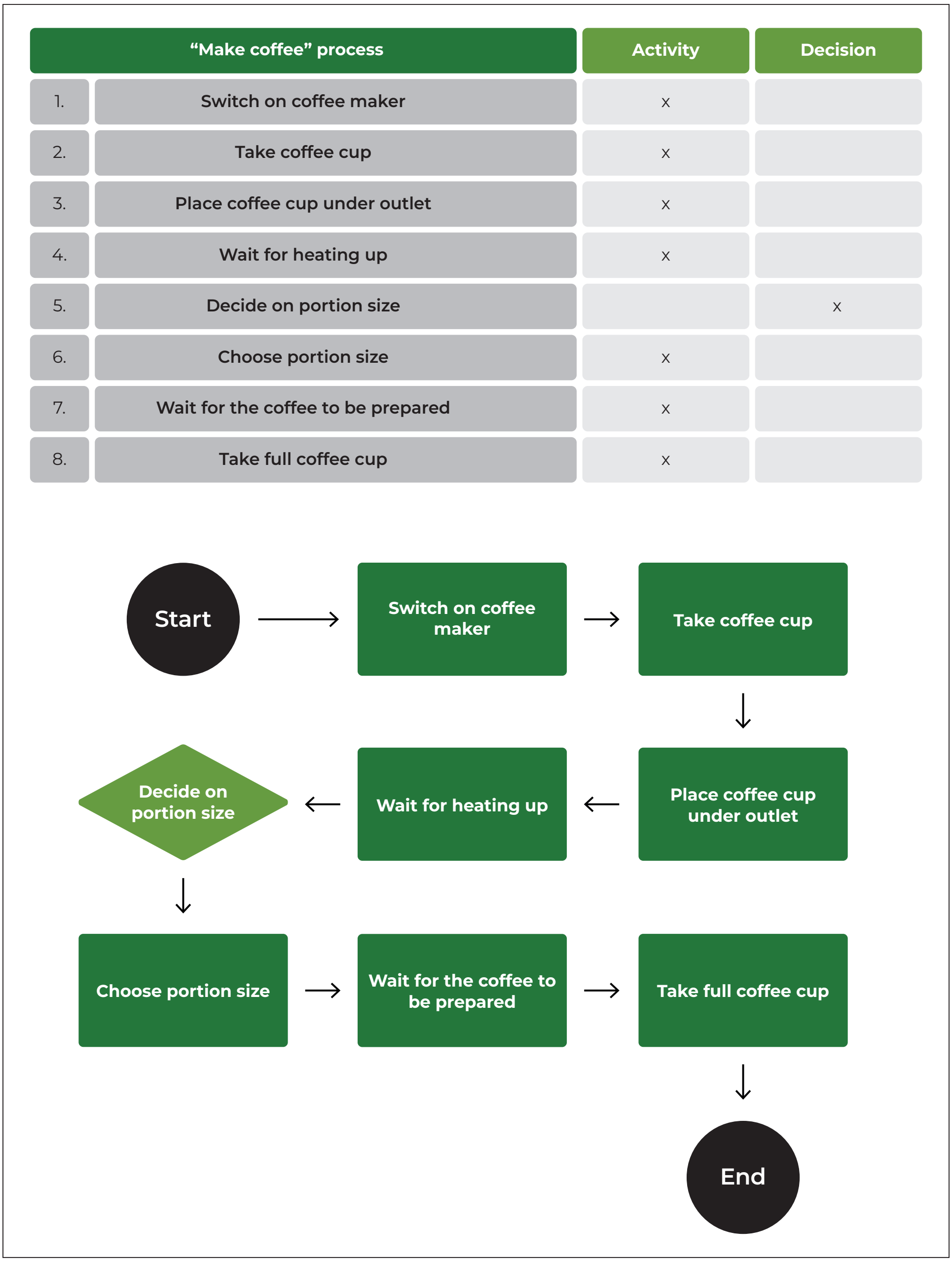
With the help of flowcharts or tables, you can then look for common patterns. We typically ask the following questions:

- Which parts of the task always run the same?
- Which task blocks always appear in a specific order to other task blocks?
- Are there parts of the task that only occur with individual persons? If so, what distinguishes this person from the other persons?
- Which parts of the task occur for all people?

You'll use the answers to these questions to perform the following tasks:

- Identify subtask blocks that are fixed
- Record fixed sequences between sub-task blocks
- Distinguish between necessary and optional parts of the task
- Find references to different user groups

In the best case, you can clearly say afterward which subtasks are compulsorily part of the task and in which order these steps are run through.



“Make coffee” process		Activity	Decision
1.	Switch on coffee maker	x	
2.	Take coffee cup	x	
3.	Place coffee cup under outlet	x	
4.	Wait for heating up	x	
5.	Decide on portion size		x
6.	Choose portion size	x	
7.	Wait for the coffee to be prepared	x	
8.	Take full coffee cup	x	

Start
Switch on coffee maker
Take coffee cup
Place coffee cup under outlet
Wait for heating up
Decide on portion size
Choose portion size
Wait for the coffee to be prepared
Take full coffee cup
End

Figure 9.20 Two different ways of presenting the task analysis: as a list or table (top) and as a flowchart (bottom).

Notice how, in our coffee example and the definition of *goal* (“What do you want to achieve?”) and *task* (“What is the current way to do this?”), an observation—and thus the task analysis—is strongly tied to the existing equipment and on-site processes as a first step. Suppose you intend to develop an entirely new solution. In this case, you must detach yourself from the concrete implementation at this point and abstract the task analysis a bit further.

An *abstracted version of the task* is usually shorter, involves fewer steps, and often includes only the decisions since the actions are tied to the implementation. Thus, at the highest level of abstraction, the task of "making coffee" could be shrunk down to the following steps:

1. Request coffee
2. Decide on portion size
3. Communicate portion size request
4. Wait for coffee

What the user must do to get the coffee again depends heavily on the implementation. As you can see, this abstracted form is suddenly transferable to different solution paths. Using the same steps, you could also order your drink in a regular café and would answer the question "cup or pot?"

Which level of abstraction is relevant for your project and your approach depends very much on your plan and how innovative your solution may be. Who says that you must drink coffee from a cup? You may remember the *Senseo* brand, which created differentiation with this question by including a glass with its machines instead of a cup, as shown in Figure 9.21. Another implementation can be "*pocket coffee*," which is a chocolate praline filled with a cold espresso shot marketed by the popular chocolate company Ferrero. This product erases waiting times, no one needs to pay attention to temperatures, and the individually wrapped chocolates mean caffeine intakes can be freely controlled.

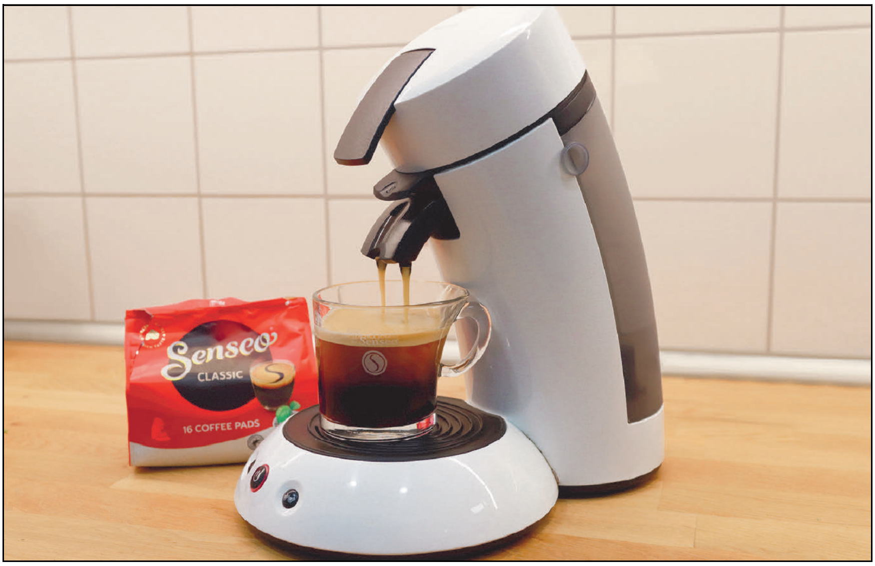


Figure 9.21 Glass instead of cup: a Senseo coffee machine from Philips.

Besides listing subtasks, two steps are beneficial for the context of use analysis and deriving usage requirements. These two steps are the addition of required inputs and the addition of created outputs:

- **Required inputs**
  What does the user need to execute this substep of the task successfully? This required input can be information, a material, an object, energy, and more. For example, a cup must be taken and placed under the nozzle.
- **Created output**
  What is present at the end of this subtask? This output could be, for example, a state (the device is switched on); a partial product (heated water, ground coffee); or a final product (finished coffee in the cup).

In general, you need complete inputs to ensure your eventual solution can be used effectively and efficiently. Not having all the information to solve the subtask will lead to workarounds, delays, and terminations. The outputs, in turn, serve you as input for further subtasks—everything needed as input must come from somewhere. On the other hand, the outputs serve you as a check: if you produce outputs that are not used anywhere, they may be simply unnecessary and could be omitted.

You can either document the inputs and outputs as extra columns in a table (inputs on the far left, then the subtasks and outputs on the far right) or attach them to the flowchart as small graphics in different colors, as shown in Figure 9.22.

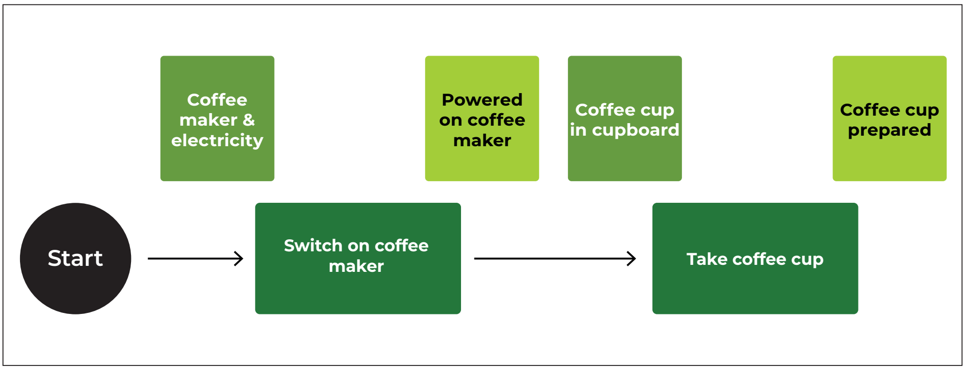


Figure 9.22 Flowchart of a task analysis with added inputs and outputs.

No limits exist on your imagination. The only important thing is understanding which inputs and outputs belong to which subtasks.

Advantages and Disadvantages

The question of what is essential and what is unimportant is a crucial part of the context of use analysis. The more knowledge you gather about the context of use and the more analytical you are when looking at it, the more likely you'll get to the heart of the task and thus be able to think of entirely new solutions.

We find it challenging to discuss the advantages and disadvantages of task analysis. The advantages (namely, understanding the user's process, current solutions, and deviations from it) are at the heart of the context of use analysis. Whether you document the findings as a table, a flowchart, or with sticky notes is a matter of taste. But we don't believe a reasonable way exists to create a good product for users without understanding their tasks and the information needed to do them. Task analysis is simply a tool to proceed systematically. However, you can use any other approach as long as you identify and systematically present the core of a task and its information needs.

However, this systematic approach and its detailed documentation are associated with disadvantages. Depending on the scope of the observed task, performing and documenting a task analysis can be pretty complex and cost a lot of time. Especially if you have many different solutions or solution paths, identifying the core task and the need for information is almost impossible via pure observation alone. In this case, a survey in the form of an *interview* (Section 9.3.3) or a *contextual inquiry* (Section 9.3.6) could be beneficial and serve as a successful addition to existing information. Although this approach rapidly improves your understanding of a task, it also significantly increases the effort required for evaluation and presentation.

In addition, a problem we've often seen is a lack of abstraction of the task quickly leads to staying too close to existing solutions. Just because there is a power button today does not mean that all future coffee machines will also have power buttons. Its function could just as well be solved by voice control, time control, or a proximity sensor. The systematic implementation of task analysis thus runs the risk of getting stuck in incremental innovation triggered by the documentation format. While this conformity may be desirable, it should not be conditioned by the documentation.

Tip: Choose the Right Level of Abstraction

To not predetermine a solution through documentation, choosing the correct level of abstraction for a task is necessary. However, determining what "correct" means in this context is often difficult, especially in a new field. As a recommendation, we suggest you always check whether the documentation is about a decision or an action or whether the description already refers to an interface element. Usually, you're going into too much detail if you're already talking about interface elements.

Nevertheless, we believe that analyzing actual behavior brings to light much information lost in pure surveys—simply because many users are not even fully aware of all their behaviors.

Table 9.10 shows the profile for this method.

Profile: Task Analysis	
Brief description	Analytical method of breaking down a task into its sub-steps to identify the information, materials, etc., needed to complete it.
Advantages	■ Ability to look at the task analytically and rethink the process based on that. ■ Breaking down the task clarifies what is needed and what is only necessitated by the chosen solution. ■ The systematic identification of commonalities in non-standardized processes becomes possible.
Disadvantages	■ Finding the right level of detail to perform task analysis can be difficult. ■ Can be overly oriented to existing solutions, thus preventing innovation. ■ Relatively large effort for the collection of the correct data and the evaluation, if it concerns unstandardized or less standardized processes.

Table 9.10 Profile of the task analysis method.

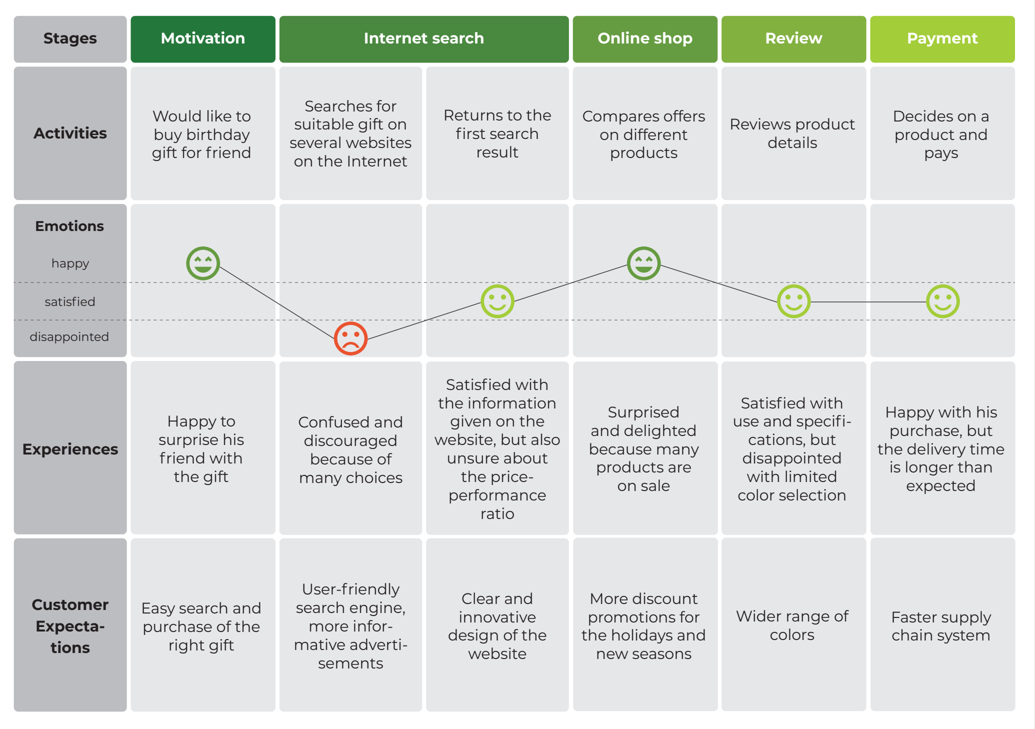
9.4.2 User or Customer Journey Mapping/Touchpoint Analysis

The *user* or *customer journey mapping* and *touchpoint analysis*, like task analysis, are not methods for obtaining new data but are a way of looking at and presenting existing data. This group of methods focuses on a graphical user or customer journey presentation.

The *user journey* refers to the various stations (e.g., website, specific subpages on the website, call center) that users pass through while using the system, product, or service. In the presentation, these stations are usually coupled with the relevant questions, tasks, and goals that the users pursue at each station, as well as with their satisfaction with the implementation of the current station. The user journey now almost always takes place via various media and points of contact, which is why touchpoint analysis is also becoming increasingly important.

In the *user journey* (we'll omit the *customer* for more effortless reading—you can decide who to look at), the individual stages of the activity are picked up and presented. In this way, the user journey is somewhat comparable to *task analysis*. However, the level of detail often differs significantly. In many cases, the user journey method divides the user's journey into individual tasks, whereas task analysis divides tasks into subtasks. Thus, task analysis allows one to virtually zoom into the user journey and look at more details and information.

User journey mapping is almost always presented in *swimlanes*, in which the activities are listed in the columns, and further information about the step is listed in the rows (the swimlanes), similar to a table. A typical example for the preparation of a user journey would be the subdivision into the areas of *journey* and *UX*. One column would represent one individual step in the journey. In the case of the UX, individual swimlanes would be used to note what goes through the user's mind during a step or how they feel about it, as shown in Figure 9.23.



Stages	**Motivation**	**Internet search**		**Online shop**	**Review**	**Payment**
Activities	Would like to buy birthday gift for friend	Searches for suitable gift on several websites on the Internet	Returns to the first search result	Compares offers on different products	Reviews product details	Decides on a product and pays
Emotions happy satisfied disappointed	☺	☹	☺	☺	☺	☺
Experiences	Happy to surprise his friend with the gift	Confused and discouraged because of many choices	Satisfied with the information given on the website, but also unsure about the price-performance ratio	Surprised and delighted because many products are on sale	Satisfied with use and specifications, but disappointed with limited color selection	Happy with his purchase, but the delivery time is longer than expected
Customer Expectations	Easy search and purchase of the right gift	User-friendly search engine, more informative advertisements	Clear and innovative design of the website	More discount promotions for the holidays and new seasons	Wider range of colors	Faster supply chain system

Figure 9.23 A user journey map. A full PDF of this image is available for download at www.rheinwerk-computing.com/5778 under the product supplements.

As with many other usability and UX design methods, no *one* right way exists here and there are many possible extensions. Jerry Angrave (2020), for example, adds the experience of the development team and suitable parameters to the scheme to track the individual steps.

In some cases, user journey maps are also used to show relationships between different tasks, for instance, general relationships where different processes are picked up in a journey map. This relationship is usually represented by arrows between the different tasks.

Think, for example, of the execution of tasks in a task analysis that you might observe in different people during execution. The situation is similar here, with entry points or jumping-off points that can be represented in a journey map. Some individuals start with task A and then jump to B. But some also start at B and come back to A later.

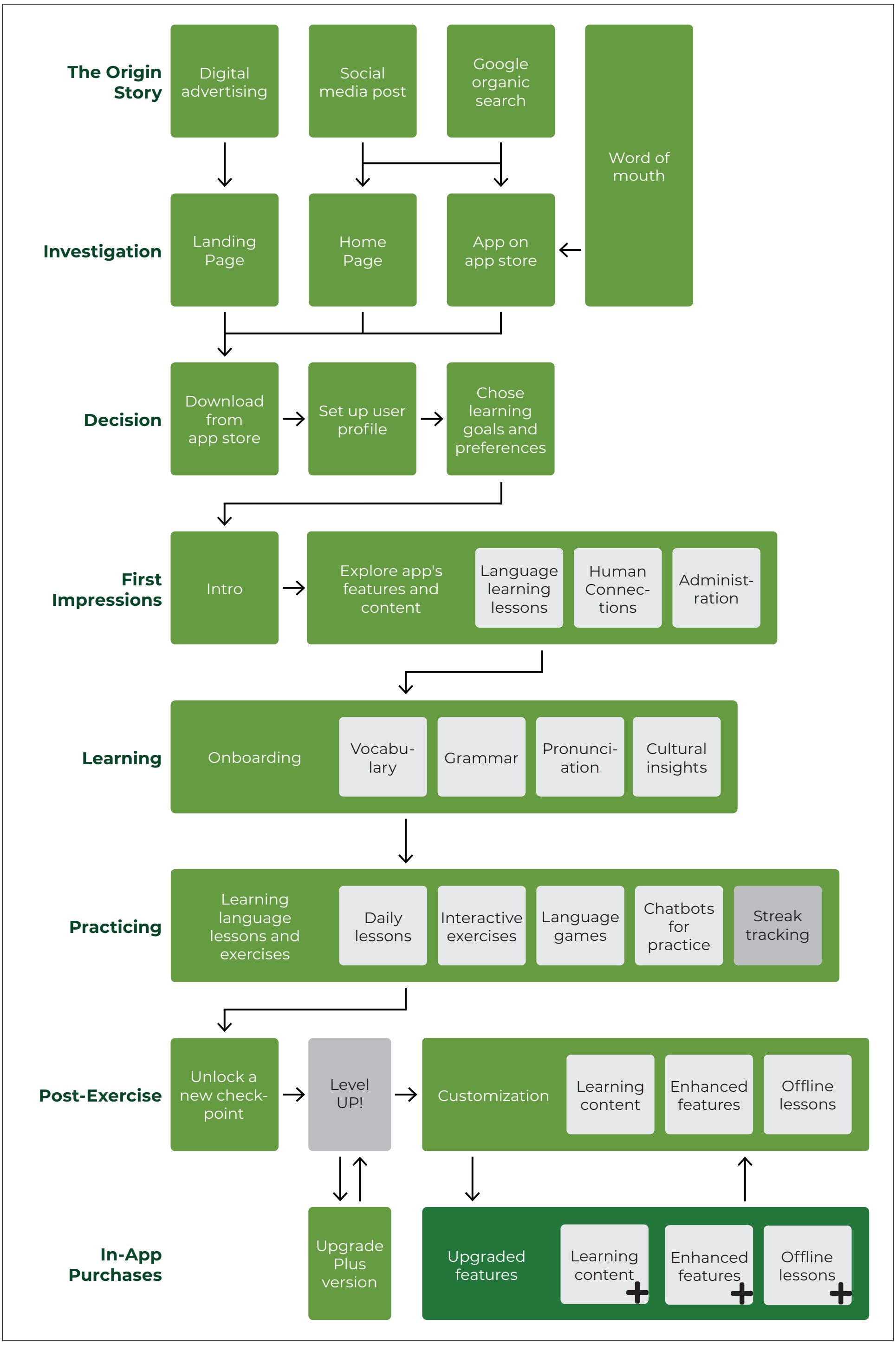

The Origin Story
Digital advertising
Social media post
Google organic search
Word of mouth
Investigation
Landing Page
Home Page
App on app store
Decision
Download from app store
Set up user profile
Chose learning goals and preferences
First Impressions
Intro
Explore app's features and content
Language learning lessons
Human Connec-tions
Administ-ration
Learning
Onboarding
Vocabu-lary
Grammar
Pronunci-ation
Cultural insights
Practicing
Learning language lessons and exercises
Daily lessons
Interactive exercises
Language games
Chatbots for practice
Streak tracking
Post-Exercise
Unlock a new check-point
Level UP!
Customization
Learning content
Enhanced features
Offline lessons
In-App Purchases
Upgrade Plus version
Upgraded features
Learning content
Enhanced features
Offline lessons

Figure 9.24 User journey map example.

For a better understanding, we have included an example that shows a journey map with these connections, as shown in Figure 9.24. In this case, a user journey is shown in which people are asked about their path to a language learning app. For the origin story, for example, various entry points (digital advertising, social media posts, Google Search, word of mouth) were used. However, the users landed at a different point depending on the entry point (landing page versus homepage versus app store page). The possible paths of the users are represented by arrows, which can run in different directions but can also meet again.

Suppose the focus is instead on the journey analysis according to the different contact points. In this case, the swimlanes primarily represent the different media and points of contact or touchpoints. For example, one swimlane could represent a company's website, another could represent its flagship store, and a third could represent its telephone customer support.

After all touchpoints have been plotted, the user's task steps are distributed across the various swimlanes. In this way, media breaks in the user journey become apparent quickly and easily. *Touchpoint analysis* offers many advantages, especially for large companies, where tasks and customer experiences are experienced across many functional areas, while companies themselves may think in terms of individual tasks and products. A fragmented UX might result, as all areas take steps to optimize the user journey, but these steps are not necessarily well coordinated.

Sometimes, similar effects arise because the touchpoints change significantly over time, resulting in an imbalance. Touchpoint analysis may not yet offer a solution, but at least it represents an analysis tool that clearly shows the problem.

No matter what type of journey mapping you use, we recommend always considering the actual use of your system, product, or service; the period before the decision to buy or use occurs; and the period after when the usage is reflected. Reflection of usage is not likely to be present in all products.

You can use a backward view exceptionally well in change situations, that is, when users part with a solution to use a new one. Strictly speaking, you're again on the way to the purchase or usage decision. Especially in the context of use analysis, understanding and depicting the interrelationships is vitally important. Therefore, looking beyond the mere use of your solution makes sense.

Advantages and Disadvantages

User journey maps are an easy way to graphically process different types of information and show their progression in the task context. They are excellent for discussion since the graphic processing and clear subdivision make the status visible to everyone. They also work well at higher organizational and macro levels, for which a task analysis is too small scale and use centered.

Since journey mapping is a group of methods and ways of presentation, the content is freely selectable and can be easily adapted to the objective and "flight level."

However, this flexibility is also a weakness of the method cluster. In most cases, it is not enough to say: "We're making a user journey map!" to ensure that expectations and results match. In this context, we strongly recommend clarifying in advance what should be the focus of consideration. Theoretically, all information can be integrated into a single map down to the task analysis results. However, comprehensibility and clarity then become problematic. Our clear recommendation: Decide on a focus for the analysis and then optimize your presentation accordingly.

Table 9.11 shows the profile for this method.

Profile: User or Customer Journey Mapping/Touchpoint Analysis	
Brief description	A representation method that can be used to prepare the data collected during the context of use analysis to easily visualize and understand the different contact points or the user's path through the solutions.
Advantages	■ The graphical presentation makes it easier to capture the data. ■ This method provides a quick overview of the data collected, making it a sound basis for discussion. ■ Depending on the content, it provides a mixture of objective and subjective data.
Disadvantages	■ No fixed display schemes so each display can look different. ■ Representations tend to simplify to keep the amount of data manageable.

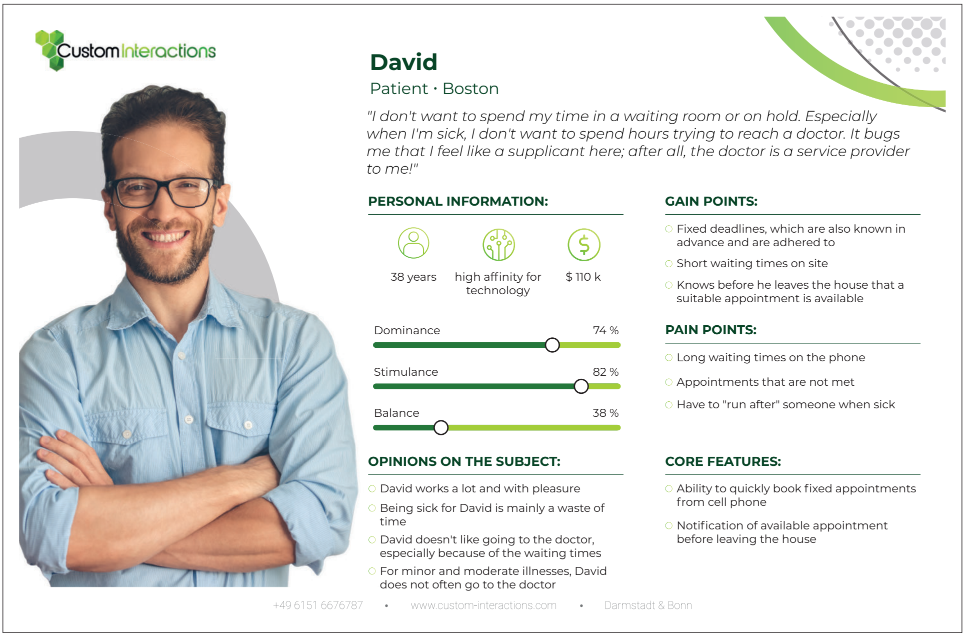
Table 9.11 Profile of the user or customer journey mapping/touchpoint analysis method.

9.4.3 Personas and Empathy Maps

Another method for data preparation is the use of personas. Personas are prototypical representations of typical user or customer groups. A persona is usually created in such a way that the user group (rather abstractly, as a group of people) is represented as a person (a concrete fictitious person with the characteristics of the group).

The time to create a new persona is whenever the data shows that the customer/user group differs based on their characteristics (which are essential for the objective and task) or based on the objectives and tasks themselves. The persona usually includes a picture and the person's name to make it easier to include the persona in discussions, as shown in Figure 9.25.

"Including the persona in the discussion" is an important clause, which we'll discuss in more detail in this section. At this point, however, we want to anticipate why personas should be created: Personas make development more straightforward. People find it much easier to think about whether a particular person might like something than whether a group might like something. Personas are a form of representation that projects abstract concepts onto a fictional person, who is then integrated into your development team.



Custom Interactions

David

Patient · Boston

"I don't want to spend my time in a waiting room or on hold. Especially when I'm sick, I don't want to spend hours trying to reach a doctor. It bugs me that I feel like a supplicant here; after all, the doctor is a service provider to me!"

PERSONAL INFORMATION:

38 years

high affinity for technology

$ 110 k

Dominance 74 %

Stimulance 82 %

Balance 38 %

OPINIONS ON THE SUBJECT:

- David works a lot and with pleasure
- Being sick for David is mainly a waste of time
- David doesn't like going to the doctor, especially because of the waiting times
- For minor and moderate illnesses, David does not often go to the doctor

GAIN POINTS:

- Fixed deadlines, which are also known in advance and are adhered to
- Short waiting times on site
- Knows before he leaves the house that a suitable appointment is available

PAIN POINTS:

- Long waiting times on the phone
- Appointments that are not met
- Have to "run after" someone when sick

CORE FEATURES:

- Ability to quickly book fixed appointments from cell phone
- Notification of available appointment before leaving the house

+49 6151 6676787 • www.custom-interactions.com • Darmstadt & Bonn

Figure 9.25 A persona for a Healthyfy target group.

Note

Full PDF versions of Figure 9.25 through Figure 9.35 are available for download at *www.rheinwerk-computing.com/5778* under the product supplements.

In addition to the photo and the name, the tasks, and the goals (the jobs-to-be-done) of the target group, you can usually add what frustrations (*pain points*) the person is currently facing in completing their jobs and what pleasures (*gain points*) they are currently deriving from doing them.

With personas—as with all other representation methods presented so far—there is no one right way to represent the content and no consensus on the necessary content exists beyond what we've said so far. The forms of representation are as diverse as the content and its use cases. A *persona should make it as easy as possible to keep track of the core information about your target audience*. Through the form of presentation, the persona should also encourage you to think about them and "let them have their say" in conversations, meetings, and discussion groups.

Very few companies, even very few products, have only one persona, and thus many products use prioritization to distinguish between various personas. The *prioritization of the personas* can be chosen based on various criteria, such as the following:

- Group size
- Group purchasing power
- The speed with which the customer group can be penetrated
- Influence on purchase/use decisions of other target groups

Which prioritization is right for your system, product, or service depends heavily on your business model, stage of development, and competitive situation. For most use cases, we recommend focusing on the most significant target groups or those with the highest purchasing power and only switching to the other suggested prioritizations for a product with a first-mover advantage. *First-mover advantage* is the advantage that the first player active in the market secures over following players (e.g., in social networks that thrive on the number of participants).

However, decisions about prioritization have little to do with classic usability and UX design but are instead located in the area of corporate strategy. The background for this recommendation is that the effort for product development is the same in almost all target groups (provided that the scope of the expected functions does not differ significantly). Thus, the cost of development remains almost the same, but the market for sales or the purchasing power of the market changes in your favor, which makes refinancing easier.

The Creation and Presentation of Personas

When creating personas, we usually proceed in two separate steps:

1. We analyze data from interviews and other qualitative methods to develop an initial version of the persona.
2. We verify the personas—if possible—using quantitative data.

As you can see from our approach, we strongly advocate building personas solely on facts. We often hear the recommendation to create personas in a workshop based on the knowledge available in the company. This can be one approach. If you want to follow this approach, make sure this workshop is facilitated by a person who has a firm grasp of this task and is willing to “go toe-to-toe” with all stakeholders present.

Moderation then includes questioning whether each piece of information recorded in the persona is genuine data-based knowledge, is an assumption, or is a gut feeling. Let that sink in: This categorization applies to every single piece of information. If you’re not careful at this point, you can easily end up with stereotypical personas or preconceptions in the participants’ minds. An example of such a stereotype might be “the introverted programmer” or the “stressed and overwhelmed single parent.” Instead, use the information from the interviews and develop personas that correspond to the categorizations you found in the analysis of the interviews.

To make your persona as meaningful as possible, we recommend that you only include information in the persona that is related to the use of your solution. Thus, if you found out that 25% of the users in the surveyed group practice magic as a hobby, then you should only include this hobby if it's a relevant connection and not just a coincidence. You can check this connection, for example, by asking what magic has to do with the solution or by looking for people with similar characteristics who don't do magic or who also do magic but still have no need for your solution.

A persona is an excellent tool to focus on user goals again and again transparently. But for this approach to work in the long term, you need data-driven personas that demonstrably reflect the relevant customer groups and their characteristics, goals, and tasks.

We recommend that you proceed with this data-driven development in two stages. The first stage is the *derivation of the persona based on the qualitative interview results* described above (analysis of data, selection of commonalities, identification of relevant information, and transfer to persona). In a perfect world, the second step is to *verify the persona using quantitative data*. In this process, you have defined in your persona which characteristics, tasks, and expectations a particular group should have, and which may not. Suppose you now design a questionnaire that asks for these characteristics, tasks, and expectations and send this questionnaire to people with matching selection criteria (e.g., age or occupation). In this case, this method allows you to do two things:

- You can check whether there are people in the group to whom this combination of data applies.
- You can check if this combination of data helps you distinguish your users from others.

If you don't find any people who have this combination, then you know probably a problem arose with the composition of the persona. If so, you can at least get a rough sense of how often that happens. Additionally, you'll get information about how your searched users differ from others. If you cannot find a difference from others based on these criteria (they all belong), then you may not yet have found the decisive criteria for defining the "right" personas since only a few product solutions should be relevant for everyone.

Pitfall: Personas without Data

If not all information can be substantiated with data, this is not bad as a first step. We call these parts or personas not supported with data *proto-personas*. The next step is to understand a proto-persona's individual information as hypotheses and back them up with data, for example, through targeted interviews or observations of a specific target group where knowledge gaps still exist. Only then can you move on because, if you don't fact check, you'll most likely cast doubt over using the persona method.

Unfortunately, we have often witnessed how information was integrated into a persona that lacked any basis in the data. Depending on how obvious this problem is, you'll either have problems with the credibility of the personas directly when they are presented or, at the end, when it becomes apparent that the content does not match real user needs. You'll later hear in conversations: "Persona? No, absolutely not. We've tried that. Doesn't work for us!" which would be a great pity.

The clean derivation, transparent presentation, and introduction of personas are crucial for their success. Since the development of personas involves a relatively high level of effort and many costs, many companies also spend a great deal of effort following the development to have *personas introduced*. We introduced the data collection methods earlier in this chapter; what is new in this section is the fact that the results of the work are widely publicized. Over the years, we have been privileged to observe many different approaches as companies try to ensure that knowledge about personas does not end up in drawers but in the everyday lives of project teams and employees.

Promising approaches that we were able to observe include the following:

- Centralized introduction of personas by management to emphasize their importance, with simultaneous provision of persona fact sheets to all affected individuals
- The introduction of the personas utilizing roleplaying sessions in which the personas "performed themselves" (and the subsequent distribution of fact sheets)
- Presenting the personas via questions to the creators involved (and distributing fact sheets)
- Having life-size personas printed and placed in meeting rooms with names and info posters (and then distributing fact sheets)
- Painting/printing the company walls and hallways with key personas and their information (and then distributing fact sheets)

As you can see, the possibilities are many. However, all the companies we know that successfully use personas manage to do so by getting the persona in the room and, in some way, ensuring that the personas are known to everyone in the company. This means not just to the people who happened to be present at the introduction but also to those who later became part of the company. Thus, creating personas is only the first step in making them effective. Precisely because the effort required to create and use personas is so high, in our view, it makes no sense to introduce personas for which you cannot even say whether this target group exists in that form. Our recommendation: Use personas only based on data and only if management accepts and supports them—ideally, even introduces them.

Empathy Maps as a Variant

A modified representation of a persona is called an *empathy map*. In this variant, the same or similar data is used during creation, but in contrast, a quasi-standard exists when creating empathy maps. In an empathy map, the user is the focal point surrounded by four or six areas, as shown in Figure 9.26.

The four core areas divide the information into the following questions:

- What does the user say to others, and what do they actually do?
- What does the user think and feel?
- What does the user see and hear from others and the environment?
- What are essential functions?

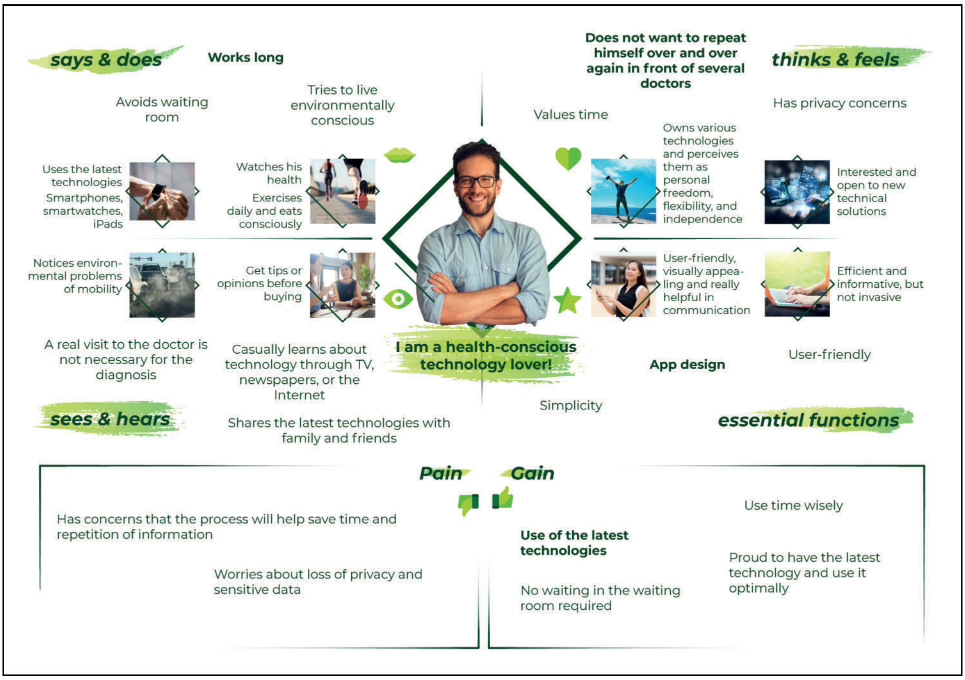

says & does
Works long
Avoids waiting room
Tries to live environmentally conscious
Uses the latest technologies Smartphones, smartwatches, iPads
Watches his health
Exercises daily and eats consciously
Notices environmental problems of mobility
Get tips or opinions before buying
Does not want to repeat himself over and over again in front of several doctors
thinks & feels
Values time
Has privacy concerns
Owns various technologies and perceives them as personal freedom, flexibility, and independence
Interested and open to new technical solutions
User-friendly, visually appealing and really helpful in communication
Efficient and informative, but not invasive
I am a health-conscious technology lover!
A real visit to the doctor is not necessary for the diagnosis
Casually learns about technology through TV, newspapers, or the Internet
sees & hears
Shares the latest technologies with family and friends
Simplicity
App design
User-friendly
essential functions
Pain
Gain
Has concerns that the process will help save time and repetition of information
Worries about loss of privacy and sensitive data
Use of the latest technologies
No waiting in the waiting room required
Use time wisely
Proud to have the latest technology and use it optimally

Figure 9.26 Example empathy map for a target group of Healthyfy.

In the extended form with six fields, this structure is supplemented by the two fields: frustration and pleasure (*pain points* and *gain points*).

We feel that the "classic" empathy map lacks focus on the user's goals, as these goals are only ever expressed indirectly. Instead, empathy maps are a good representation of the more emotional aspects. In practice, however, the separation between the fields is often unclear, and discussions can often arise about why a certain point is in one field and not another. From our point of view, however, these disagreements are not tragic as long as you ensure all relevant information is included somewhere.

You should try out which method suits you best when you can. Empathy maps are also often used in the design thinking context (see Lewrick, Link, & Leifer, *The Design Thinking Playbook*, 2018) and created in workshops with the team. As a result, they more often look "unfinished" because they consist of sticky notes and handwriting, as shown in Figure 9.27.

This unfinished look has advantages and disadvantages: On the one hand, an empathy map may be taken less seriously precisely because it looks unfinished. On the other hand, the empathy map with this appearance encourages the integration of newly gained knowledge. Especially when working classically and not remotely, an empathy map can easily be extended using sticky notes. In the meantime, however, plenty of online interaction tools enable digital sticky notes and thus force a similarly iterative and agile procedure.

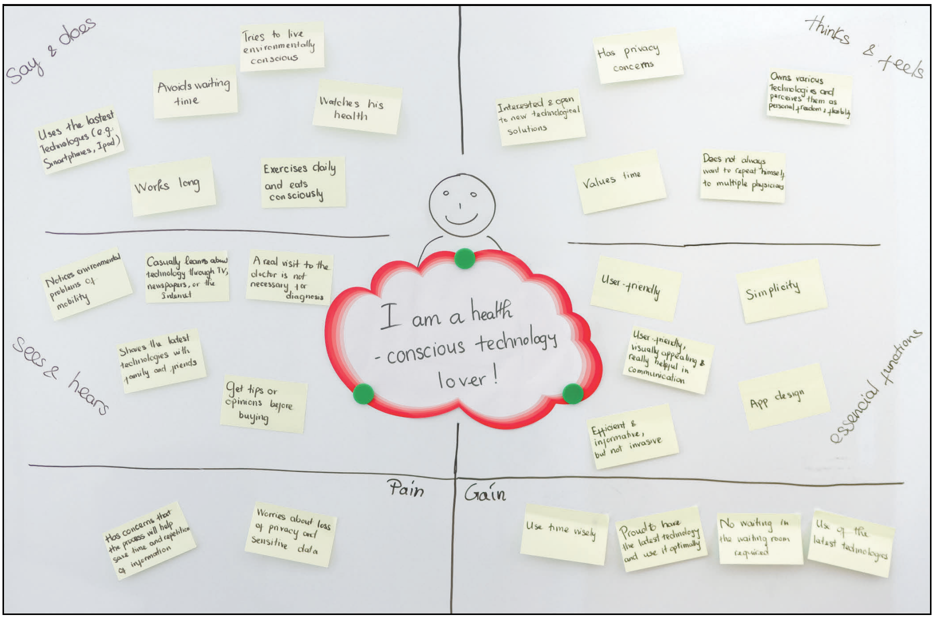

Say & does
Tries to live environmentally conscious
Avoids waiting time
Watches his health
Uses the lastest Technologues (e.g.: Smartphones, Ipad)
Works long
Exercises daily and eats consciously
thinks & feels
Has privacy concerns
Owns various technologies and perceives them as personal freedom & flexibility
Interested & open to new technological solutions
Values time
Does not always want to repeat himself to multiple physicians
Notices environmental problems of mobility
Casually learns about technology through TV, newspapers, or the Internet
A real visit to the doctor is not necessary for diagnosis
Sees & hears
Shares the latest technologies with family and friends
Get tips or opinions before buying
I am a health - conscious technology lover!
User-friendly
Simplicity
User-friendly, visually appealing & really helpful in communication
App design
Efficient & informative, but not invasive
essencial functions
Pain
Gain
Has concerns that the process will help save time and repetition of information
Worries about loss of privacy and sensitive data
Use time wisely
Proud to have the latest technology and use it optimally
No waiting in the waiting room required
Use of the latest technologies

Figure 9.27 Example empathy map for a Healthyfy target group, after a workshop.

Advantages and Disadvantages

The advantages of personas and empathy maps lie in their concise and clear summary of the core characteristics of the target groups. A good understanding of personas and empathy maps makes it possible to bring the users into the room, even if they are not there. This advantage is offset by the relatively high effort required to develop accurate personas, to make them known in your company, and to keep them alive permanently.

In our view, however, there is no way around cleanly defining your target group at least once. So that you don't end up with a generic and not useful form of target group description, an essential step is also considering the context in which a person moves in the persona's description or the empathy map. The person you're talking about has a task they want to fulfill within a specific framework. Only if you think along with this framework will you arrive at a description that will help you further. But probably the biggest drawback is that personas and empathy maps are often created based on existing experience rather than actual data. Thus, errors and biases can creep in, undermining the credibility of your personas.

Table 9.12 shows the profile for this method.

Profile: Personas and Empathy Maps	
Brief description	Summary and graphical presentation of the core characteristics of the user groups so that a representative representation exists for each target group.
Advantages	■ Clear presentation of the core properties of the user groups. ■ Personas help consider the users' needs even when not on-site.
Disadvantages	■ Without a definition of the framework, usually relatively worthless. ■ Personas and empathy maps are often defined based on experience rather than actual data—leading to errors and bias. ■ High effort to develop "real" personas based on data and to make them known. ■ Personas must be actively kept alive (update, but also deployment).

Table 9.12 Profile of the personas and empathy maps method.

9.5 Practical Examples for Context Analysis

In this final section, we present possible ways our three example companies conduct a context of use analysis. Note that we're presenting the process in an abbreviated form in this section and that countless other approaches for these companies could exist.

Most importantly, in summarizing insights into personas, we do not distinguish between proto-personas (assumptions and no knowledge) and actual personas. We assume that companies have either verified the assumptions for each item on a persona or know they still need to do so.

Let's start with the examples!

9.5.1 Context Analysis at moveHUB

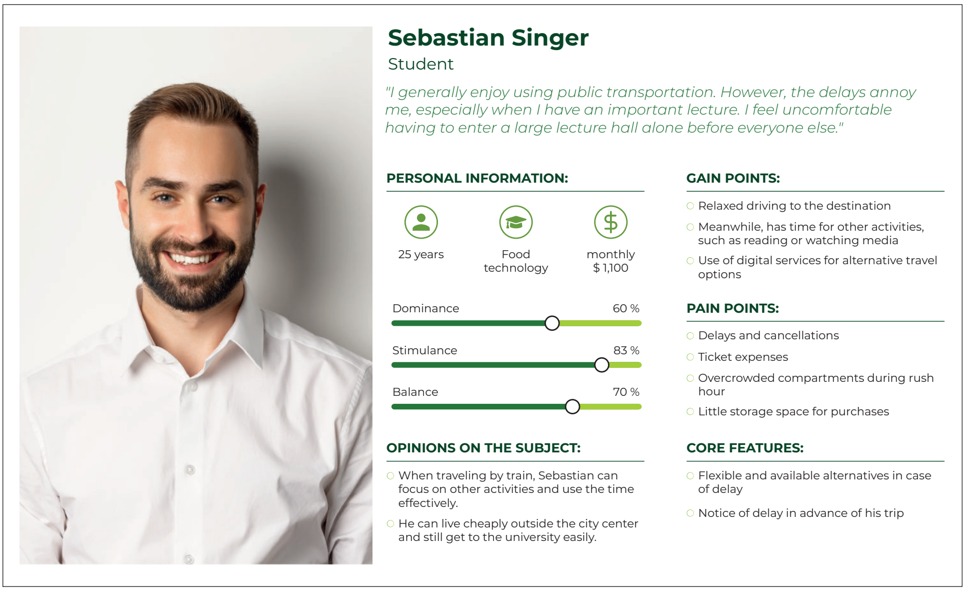
Matthew, Emma, and Anthony are still at the very beginning with *moveHUB*. Based on their own experiences, the three have discovered a problem they believe many others share. To test whether this is the case, the three decide to start working in the rural region around Austin and begin with field observations at train stations, cab stands, and bus stops. In doing so, Matthew, Emma, and Anthony focus their attention on the following questions:

- What are the busiest times?
- Do people switch between different offerings?
- To which age group do typical commuters belong?

Their initial observations reveal that most commuters heading toward the city are in their 20s to mid-30s, with a few exceptions toward the top. The three approach some commuters and conduct brief *on-site interviews*. These interviews show that many commuters are still students or are new to a job and do not have their own families. They often use the commute time for other activities, such as reading, “Netflixing,” or sleeping.

After these interviews, Matthew, Emma, and Anthony follow some of their interviewees using *shadowing* to better understand their commutes and not limit their knowledge to the starting and transfer stations. During the shadowing, the three notice that large parts of the journey are pretty relaxed for the people they observe but that delays can lead to spikes in stress. Matthew, Emma, and Anthony frequently observe how the persons regularly tear their smartphones out of their pockets and open several apps and websites one after the other in a panic to find suitable alternative transfer offers. In addition, Matthew, Emma, and Anthony discover a second group of university or work commuters who are completely stressed out by the commute and possible delays. For these mainly female commuters, stress levels increase especially when delays occur on the way home and in the evening, resulting in random waiting times at train stations and outside.

In the follow-up, our three founders take the time to look at all the data in detail once again and develop the persona on this basis, *Sebastian “The Student” Singer*, who has a commute with at least one transfer between public transportation on his way to college or his first job, as shown in Figure 9.28.



Sebastian Singer

Student

"I generally enjoy using public transportation. However, the delays annoy me, especially when I have an important lecture. I feel uncomfortable having to enter a large lecture hall alone before everyone else."

PERSONAL INFORMATION:

25 years

Food technology

monthly $ 1,100

Dominance 60 %

Stimulance 83 %

Balance 70 %

OPINIONS ON THE SUBJECT:

○ When traveling by train, Sebastian can focus on other activities and use the time effectively.

○ He can live cheaply outside the city center and still get to the university easily.

GAIN POINTS:

○ Relaxed driving to the destination

○ Meanwhile, has time for other activities, such as reading or watching media

○ Use of digital services for alternative travel options

PAIN POINTS:

○ Delays and cancellations

○ Ticket expenses

○ Overcrowded compartments during rush hour

○ Little storage space for purchases

CORE FEATURES:

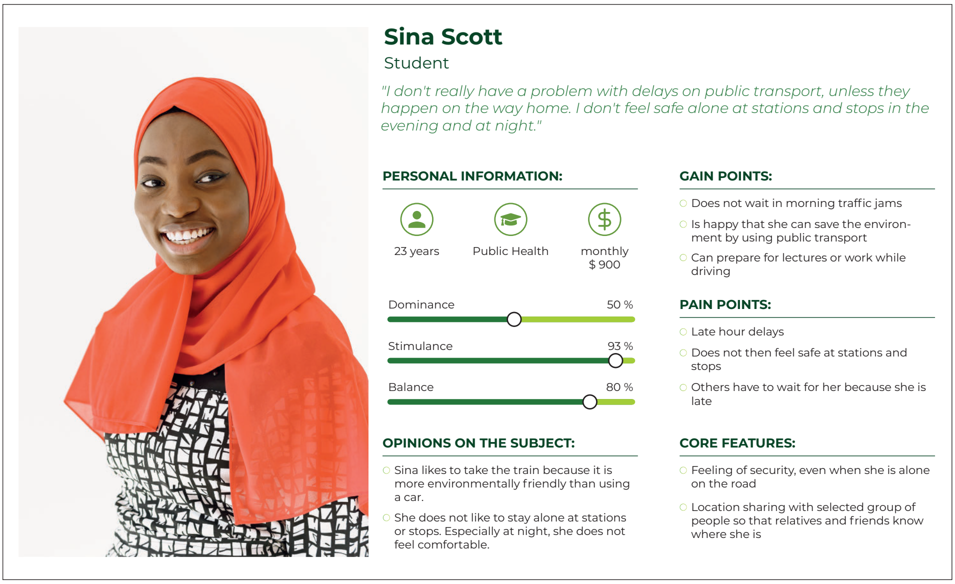
○ Flexible and available alternatives in case of delay

○ Notice of delay in advance of his trip

Figure 9.28 Sebastian “The Student” Singer persona for moveHUB.

Sebastian's main task is to be at a specific place on time to avoid getting into unpleasant situations publicly (e.g., "entering a large lecture hall alone before everyone else" or "being late for a team meeting with the boss").

Next to Sebastian Singer, we have another student persona *Sina "The Student" Scott*, as shown in Figure 9.29. Sina also commutes regularly with at least one change to university or work. Unlike Sebastian, however, Sina has no problem being a little late due to delays.



Sina Scott

Student

"I don't really have a problem with delays on public transport, unless they happen on the way home. I don't feel safe alone at stations and stops in the evening and at night."

PERSONAL INFORMATION:

23 years

Public Health

monthly $ 900

Dominance 50 %

Stimulance 93 %

Balance 80 %

OPINIONS ON THE SUBJECT:

- Sina likes to take the train because it is more environmentally friendly than using a car.
- She does not like to stay alone at stations or stops. Especially at night, she does not feel comfortable.

GAIN POINTS:

- Does not wait in morning traffic jams
- Is happy that she can save the environment by using public transport
- Can prepare for lectures or work while driving

PAIN POINTS:

- Late hour delays
- Does not then feel safe at stations and stops
- Others have to wait for her because she is late

CORE FEATURES:

- Feeling of security, even when she is alone on the road
- Location sharing with selected group of people so that relatives and friends know where she is

Figure 9.29 Sina "The Student" Scott persona for moveHUB.

On the other hand, Sina's stress level increases significantly when delays occur on the way home, resulting in longer waiting times at train stations or stops. The later the journey home takes place and the longer the expected waiting time, the more energetic Sina is in her search for a suitable alternative. Sina's goal is to return home safely without any unpleasant encounters. In addition to the safety factor, Sina often has meetings with friends in the evenings. Since the friends often meet at Sina's home, she is uncomfortable having her friends wait outside a locked door.

The founding team faces quite a challenge with the two personas. In fact, at first glance, both personas have comparable starting situations and similar challenges. Still, neither Matthew, Emma, nor Anthony is sure whether both personas would be satisfied with the same solution. More work still needs to be done here!

9.5.2 Context Analysis at BusinessBooster

Since SME IT Inc. already has an existing product, Markus Meier decides to start his context analysis using the existing data.

As a first step, Markus gets an analysis of sales figures, sales cycles, demonstration dates, number of test access events, the rate of actual sales, and the average usage time over the last few years. Looking at the numbers, he discovers that, while the number of requests for trial access and demonstration appointments has remained relatively stable over the years, the number of sales has declined every year. Initially, SME IT Inc. could convert 100 trial accounts into 25 paying customers. At the time of the project launch, this number is down to just 3. The only good news Markus can discover in the figures is that a newly acquired customer is just as loyal to SME IT Inc. as an existing customer.

Due to the radical changes in these figures over the last few years, Markus has decided to launch two parallel initiatives:

1. He asks his customer service team to call anyone with trial access who decided not to purchase the software and briefly discuss their reasons.
2. He asks customer service to set up on-site appointments at the customer's premises with all newly acquired customers from the last two years and with long-standing existing customers. The goal: to understand what convinced customers about BusinessBooster.

When Markus examines the data obtained, a clear pattern quickly emerges: BusinessBooster's happy customers are all smaller companies with few contacts to maintain and where BusinessBooster is often the only software used in contact management and invoicing. Customers praise that only one software is needed and that creating contacts and viewing information is easy. In addition, the customer service provided by SME IT Inc. represents real added value. Nevertheless, it also emerges from the interviews that the look of the software is getting on in years and that growing companies, in particular, consider the software's functionality too limited in some places. For example, the lack of automation options for reports and data migration is criticized.

The test customers who decided against using the tool also criticized the outdated look, but most state that the software's range of functions is insufficient for their internal tasks. The larger companies, in particular, struggle with the lack of interfaces to other systems and import and export options. Automated creation and sending of reports are also not possible, which means a lot of additional manual effort that could be taken over by other tools. These dissatisfied test customers primarily wanted automated evaluation and reporting for their company's sales activities.

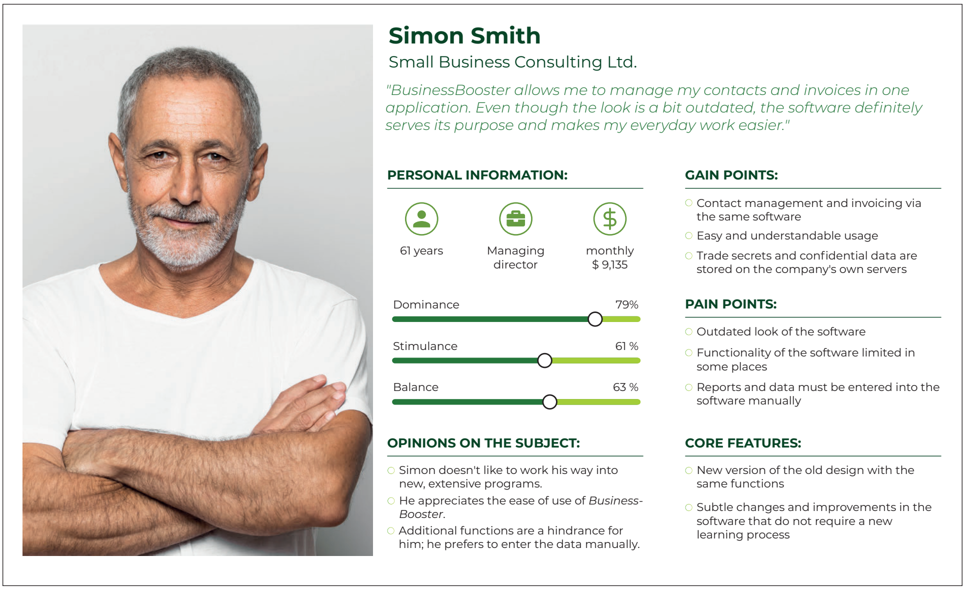
However, a second group of companies have also decided against BusinessBooster. These companies are small and very digital. They don't like that the software solution runs on the employee's PC and their central server. Because these companies are used to distributed work, these companies primarily want a web solution that employees can access from anywhere. In addition, no updates should be required to be installed on the PCs, and a central server typically does not exist in this context either.

This group also lacks options for synchronizing data from social media platforms or other tools and evaluation options for leads and sales, separated according to different funnels. Although the tool offers much of the information individually and there is also the option to filter this information, young, digital companies want a central, configurable dashboard on which all information is immediately visible.

Among existing and test customers, no company has more than 250 employees. Since SME IT Inc. has systematically collected this data, the evaluation shows that not a single corporation wanted to become a customer of the company. Markus can live well with that; after all, the name of his company is also their mission statement: boosting small businesses.

Based on these findings, Markus and his team created three personas:

- Simon "Small Business" Smith, Managing Director of Small Business Consulting Ltd.—Simon Smith represents the persona of satisfied customers, as shown in Figure 9.30.



Simon Smith

Small Business Consulting Ltd.

"BusinessBooster allows me to manage my contacts and invoices in one application. Even though the look is a bit outdated, the software definitely serves its purpose and makes my everyday work easier."

PERSONAL INFORMATION:

61 years

Managing director

monthly $ 9,135

Dominance 79%

Stimulance 61 %

Balance 63 %

GAIN POINTS:

- Contact management and invoicing via the same software
- Easy and understandable usage
- Trade secrets and confidential data are stored on the company's own servers

PAIN POINTS:

- Outdated look of the software
- Functionality of the software limited in some places
- Reports and data must be entered into the software manually

OPINIONS ON THE SUBJECT:

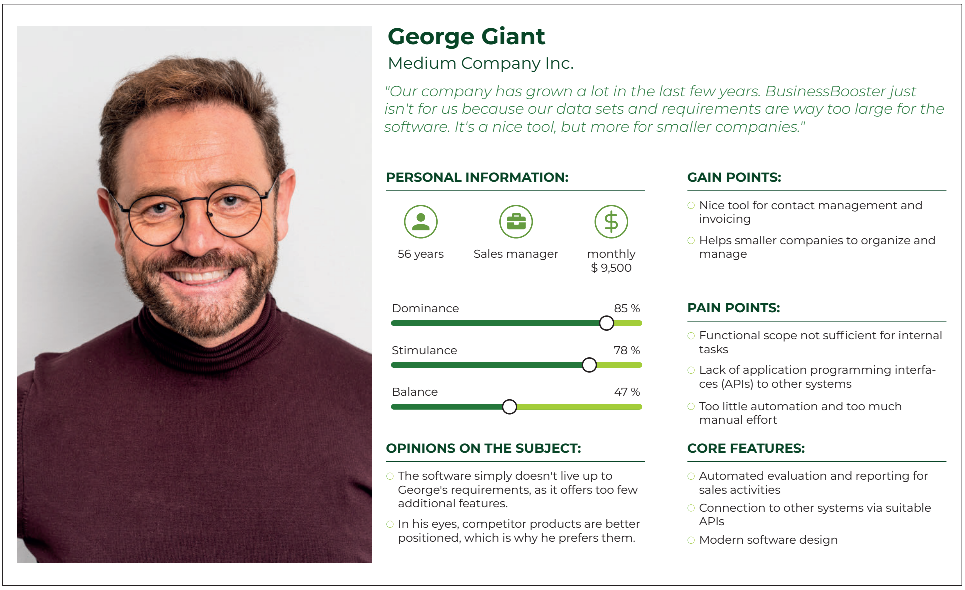
- Simon doesn't like to work his way into new, extensive programs.
- He appreciates the ease of use of *BusinessBooster.*
- Additional functions are a hindrance for him; he prefers to enter the data manually.

CORE FEATURES:

- New version of the old design with the same functions
- Subtle changes and improvements in the software that do not require a new learning process

Figure 9.30 Simon "Small Business" Smith persona for BusinessBooster.

- *George "Growth" Giant*, Sales Manager of Medium Company Inc.—as shown in Figure 9.31, George Giant represents a growing company's customer group that wants a stronger connection to other systems.



George Giant

Medium Company Inc.

"Our company has grown a lot in the last few years. BusinessBooster just isn't for us because our data sets and requirements are way too large for the software. It's a nice tool, but more for smaller companies."

PERSONAL INFORMATION:

56 years

Sales manager

monthly $ 9,500

Dominance 85 %

Stimulance 78 %

Balance 47 %

OPINIONS ON THE SUBJECT:

- The software simply doesn't live up to George's requirements, as it offers too few additional features.
- In his eyes, competitor products are better positioned, which is why he prefers them.

GAIN POINTS:

- Nice tool for contact management and invoicing
- Helps smaller companies to organize and manage

PAIN POINTS:

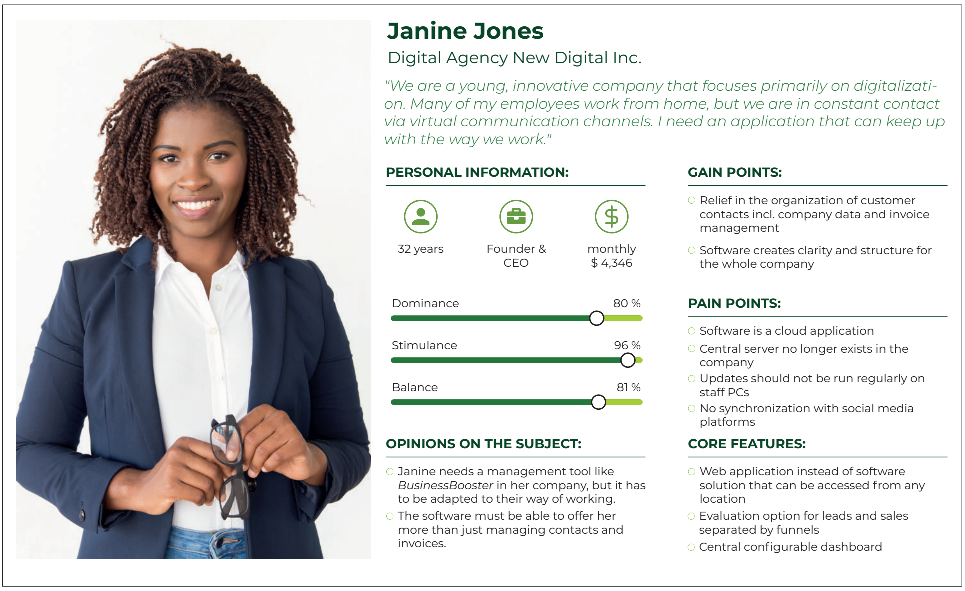
- Functional scope not sufficient for internal tasks
- Lack of application programming interfaces (APIs) to other systems
- Too little automation and too much manual effort

CORE FEATURES:

- Automated evaluation and reporting for sales activities
- Connection to other systems via suitable APIs
- Modern software design

Figure 9.31 George “Growth” Giant persona for BusinessBooster.

- *Janine “The Young Digital” Jones*, head of digital agency New Digital Inc.—Janine Jones represents all small, digital companies that sell through different funnels and work in a distributed manner, as shown in Figure 9.32.



Janine Jones

Digital Agency New Digital Inc.

"We are a young, innovative company that focuses primarily on digitalization. Many of my employees work from home, but we are in constant contact via virtual communication channels. I need an application that can keep up with the way we work."

PERSONAL INFORMATION:

32 years

Founder & CEO

monthly $ 4,346

Dominance 80 %

Stimulance 96 %

Balance 81 %

OPINIONS ON THE SUBJECT:

- Janine needs a management tool like *BusinessBooster* in her company, but it has to be adapted to their way of working.
- The software must be able to offer her more than just managing contacts and invoices.

GAIN POINTS:

- Relief in the organization of customer contacts incl. company data and invoice management
- Software creates clarity and structure for the whole company

PAIN POINTS:

- Software is a cloud application
- Central server no longer exists in the company
- Updates should not be run regularly on staff PCs
- No synchronization with social media platforms

CORE FEATURES:

- Web application instead of software solution that can be accessed from any location
- Evaluation option for leads and sales separated by funnels
- Central configurable dashboard

Figure 9.32 Janine “The Young Digital” Jones persona for BusinessBooster.

Based on strategic considerations, Markus decides that *Simon "Small Business" Smith* should be the essential persona since existing customers secure the business and should not be lost under any circumstances. Based on market size and required development scope, he also decides that the second most crucial persona is *Janine "The Young Digital" Jones*. Based on the feedback, he assumes that short decision-making paths and fast purchase cycles can be realized, which is an important criterion for SME IT Inc., especially in the current situation. *George "Growth" Giant*, meanwhile, is a persona that may not be a great fit for BusinessBooster at the moment but is a target group where their product can seek improvements.

9.5.3 Context Analysis at Healthyfy

Linda faces some major challenges building a healthcare platform that serves what's called a *two-sided market*. Healthyfy is an offering that is only exciting if both patients and doctors decide actually to use the platform. If one side is missing, its usage is not interesting for the other side.

Since Linda only has a little more than a year for her proof-of-concept, she and her team decide not to approach all physician groups in parallel but to limit themselves to specialists who are challenging to find. The first specialty the team decides on is diabetologists, who often have a regular care relationship with their patients and would not be required to be on-site, which can be seen advantages for the platform. Furthermore, the team thus circumvents the original problem by showing illnesses and injuries via the camera. Another plus point is that, because diabetes is so widespread, many potential patients could be found. The same applies to doctors.

Linda and her team decided to conduct focus groups with patients and physicians to question the concept idea. The focus will be on whether the health platform in its current form makes sense and what obstacles would speak against the use of the platform.

Linda and her team conduct the first focus group exclusively with diabetes patients. The result of the focus group shows a certain skepticism about using the app for the first contact with a doctor. In this context, patients want to get to know each other in person to decide whether they feel comfortable with the doctor. However, this does not apply to selecting a suitable doctor. In this case, users would be happy to receive assistance—for example, reviews of available physicians. For subsequent check-ups, many patients find the idea of using a health platform attractive. Above all, the idea of not having to go to the doctor in person every time, but talking about values and next steps together in a virtual appointment, is potentially a very positive experience.

A prescription for insulin available digitally would also be a good reason to use the platform. This idea is true especially for relatively young patients, around the end of their 40s. Older diabetes patients do not see much added value in the platform and would prefer close, personal support from their local doctors. Older patients also find relying on anonymous ratings when selecting a doctor somewhat off-putting. However, providing a prescription without a doctor's visit does add value for them, especially for significantly older patients already with physical limitations. Still, concerns exist about the accessibility of the prescription, as sometimes, no digital media is available.

In the second focus group, the Healthyfy team talks to physicians. There's a great deal of skepticism among them. Physicians see no advantage in using the platform, as they have enough patients. Additionally, physicians do not want to deal with another tool and invest even more time setting up technology than they currently do. Setting up and maintaining technology is a significant problem for the physicians surveyed, as existing staff do not have deep technical expertise.

These physicians also view selection and evaluation on the platform as critical and fear unqualified evaluations by laypersons. Doctors also wonder how they are protected from false ratings. Further concerns emerge regarding the protection of patients from fraudsters.

From an organizational perspective, physicians balk at the expense of maintaining two patient records and don't like having to maintain patient data separately in their own system and on the health platform.

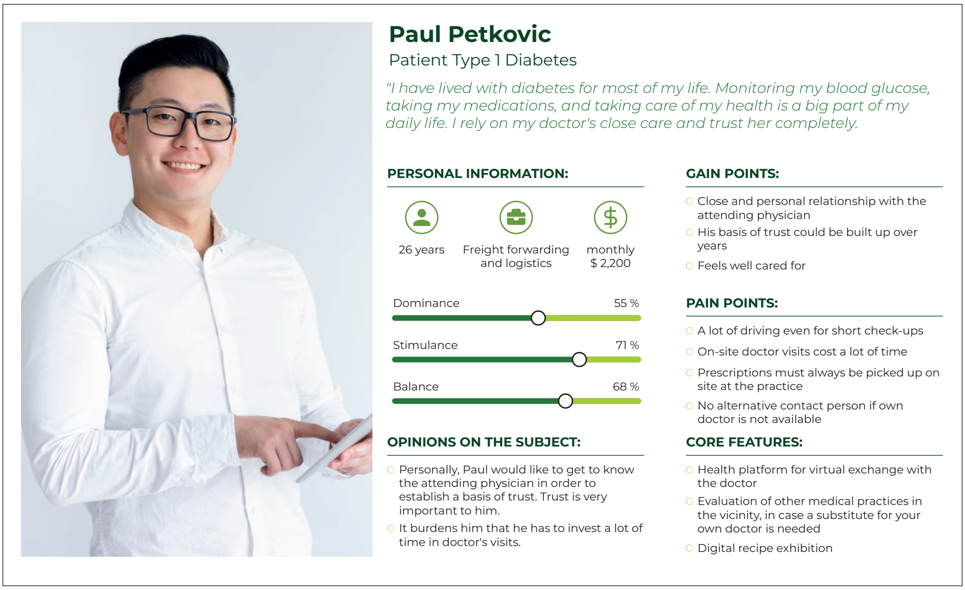
Doctors may find the idea of digitally distributing regular prescriptions exciting in itself. However, many fear they will neglect their duty of care if they have not spoken to the patient beforehand. Many physicians find indirect contact via digital channels challenging to get used to and therefore see only a few points that would speak in favor of the model.

In the focus groups, the question of what the health platform would make better or easier about their work came up repeatedly. At least in the focus group, no good answer could be found initially. But what emerges is that the need to save time clearly outweighs the physician's desire for new ways to interact with patients.

Linda and her team first feel the results are a real damper since physicians were highly skeptical of the idea. Still, the team is sure it can easily design a product for the *Paul "The Patient" Petkovic* persona that will excite that user, as shown in Figure 9.33.

For *Donna "The Diabetologist" Davis*, on the other hand, as shown in Figure 9.34, the team is a little perplexed in the first step.

The team then discusses the following sensible steps together. Everyone agrees that the patient side is easier to approach and that searching for a new specialist is the decisive moment. The team once again analyzes all the responses from the focus group conducted with patients and outlines the patient's user journey, as shown in Figure 9.35. The team feels prepared to tackle the patient solution with these results. Let's see what happens next!



Paul Petkovic

Patient Type 1 Diabetes

"I have lived with diabetes for most of my life. Monitoring my blood glucose, taking my medications, and taking care of my health is a big part of my daily life. I rely on my doctor's close care and trust her completely.

PERSONAL INFORMATION:

26 years	Freight forwarding and logistics	monthly $ 2,200

Dominance 55 %

Stimulance 71 %

Balance 68 %

OPINIONS ON THE SUBJECT:

- Personally, Paul would like to get to know the attending physician in order to establish a basis of trust. Trust is very important to him.
- It burdens him that he has to invest a lot of time in doctor's visits.

GAIN POINTS:

- Close and personal relationship with the attending physician
- His basis of trust could be built up over years
- Feels well cared for

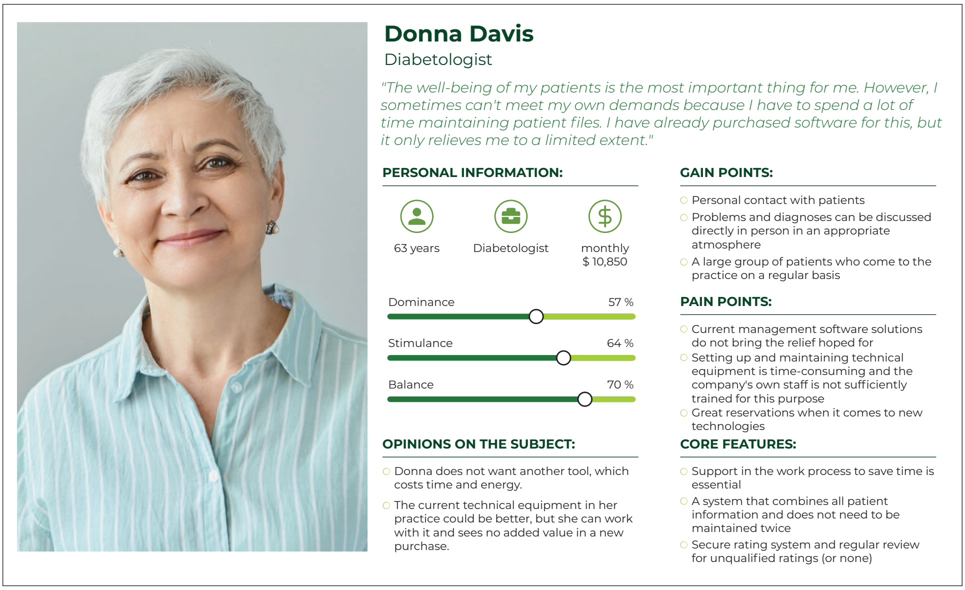
PAIN POINTS:

- A lot of driving even for short check-ups
- On-site doctor visits cost a lot of time
- Prescriptions must always be picked up on site at the practice
- No alternative contact person if own doctor is not available

CORE FEATURES:

- Health platform for virtual exchange with the doctor
- Evaluation of other medical practices in the vicinity, in case a substitute for your own doctor is needed
- Digital recipe exhibition

Figure 9.33 Paul "The Patient" Petkovic persona for Healthyfy.



Donna Davis

Diabetologist

"The well-being of my patients is the most important thing for me. However, I sometimes can't meet my own demands because I have to spend a lot of time maintaining patient files. I have already purchased software for this, but it only relieves me to a limited extent."

PERSONAL INFORMATION:

63 years	Diabetologist	monthly $ 10,850

Dominance 57 %

Stimulance 64 %

Balance 70 %

OPINIONS ON THE SUBJECT:

- Donna does not want another tool, which costs time and energy.
- The current technical equipment in her practice could be better, but she can work with it and sees no added value in a new purchase.

GAIN POINTS:

- Personal contact with patients
- Problems and diagnoses can be discussed directly in person in an appropriate atmosphere
- A large group of patients who come to the practice on a regular basis

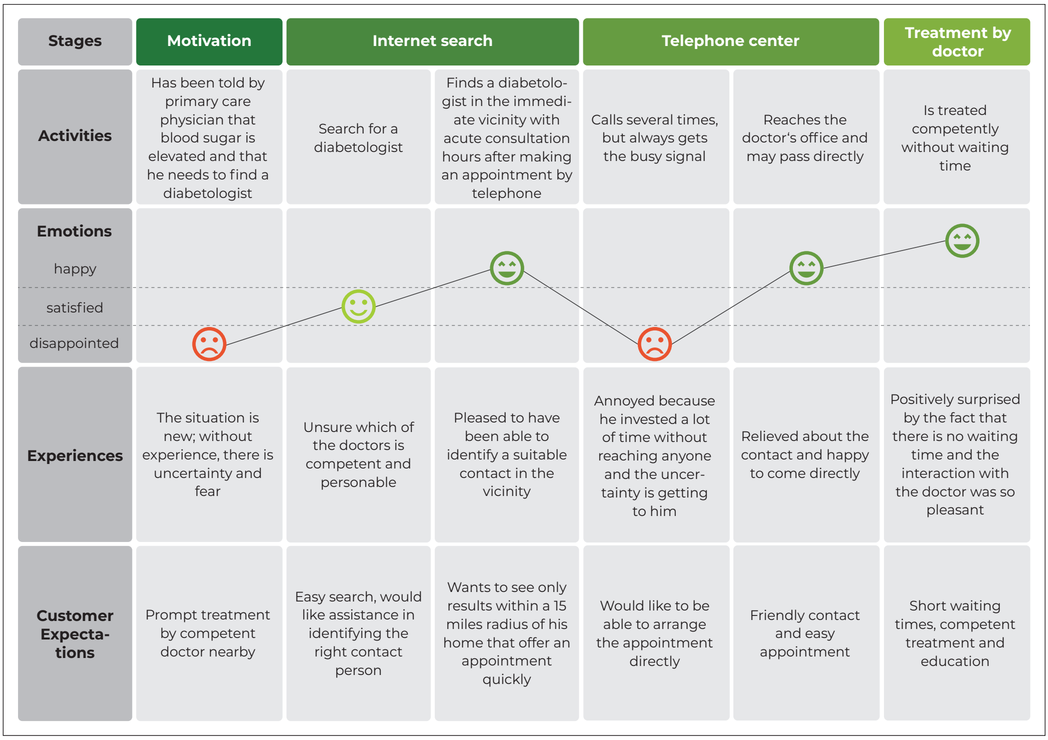
PAIN POINTS:

- Current management software solutions do not bring the relief hoped for
- Setting up and maintaining technical equipment is time-consuming and the company's own staff is not sufficiently trained for this purpose
- Great reservations when it comes to new technologies

CORE FEATURES:

- Support in the work process to save time is essential
- A system that combines all patient information and does not need to be maintained twice
- Secure rating system and regular review for unqualified ratings (or none)

Figure 9.34 Donna "The Diabetologist" Davis persona for Healthyfy.



Stages	Motivation	Internet search		Telephone center		Treatment by doctor
Activities	Has been told by primary care physician that blood sugar is elevated and that he needs to find a diabetologist	Search for a diabetologist	Finds a diabetologist in the immediate vicinity with acute consultation hours after making an appointment by telephone	Calls several times, but always gets the busy signal	Reaches the doctor's office and may pass directly	Is treated competently without waiting time
Emotions happy satisfied disappointed	☹	🙂	😆	☹	😆	😆
Experiences	The situation is new; without experience, there is uncertainty and fear	Unsure which of the doctors is competent and personable	Pleased to have been able to identify a suitable contact in the vicinity	Annoyed because he invested a lot of time without reaching anyone and the uncertainty is getting to him	Relieved about the contact and happy to come directly	Positively surprised by the fact that there is no waiting time and the interaction with the doctor was so pleasant
Customer Expectations	Prompt treatment by competent doctor nearby	Easy search, would like assistance in identifying the right contact person	Wants to see only results within a 15 miles radius of his home that offer an appointment quickly	Would like to be able to arrange the appointment directly	Friendly contact and easy appointment	Short waiting times, competent treatment and education

Figure 9.35 User journey for a patient searching for a medical specialist.

Chapter 10

Data-Driven UX Design in Detail: Specifying Usage Requirements

"Recognizing the problem is more important than recognizing the solution because the accurate representation of the problem leads to the solution."
—Albert Einstein (physicist)

In the previous chapter, you learned about many different methods that can help you better understand the context in which of your users operate better. If everything has gone according to plan, you're now in a situation where you know quite a lot and "only" have to ensure that this knowledge arrives in your product later.

But how can this inclusion be ensured? One way is through requirements. *Requirements* define what a system, a product, or a service must later achieve. In the "old world" of waterfall-based development, developments were started by collecting all requirements in a written form before technical development began.

Internal or external clients usually created this collection of requirements, whereas the development team derived the *requirements specification* from it. The *requirements specification* is a kind of project plan that shows how requirements are to be met. In conjunction with the requirements specification, usually an estimate of the time required for implementation and the resulting costs were drafted.

In projects in which hardware is developed, requirements specifications still play an essential role today. However, this procedure has been replaced mainly by agile procedures that work with what are called "backlogs" in the software industry. In a backlog, all open work tasks are collected. Additionally, time estimates are included for the individual packages. Development work is then divided into individual *sprints* in which specific tasks from the backlog are implemented. Compared to a conventional requirements specification, this procedure has an advantage in that projects can be planned in time based on the development sprints. However, the factual content can be defined flexibly as required for each sprint and adjusted later. Costs and time can still be estimated in advance based on a certain number of sprints.

This change in project planning mainly arose because it has been shown repeatedly that thoroughly estimating all requirements at the beginning of a development project is practically impossible. However, in projects that expect a fixed project plan, including content and times, this problem has repeatedly led to disappointment: either because new requirements were discovered during development, and everything became more expensive than expected or because new requirements were discovered but were not implemented at all due to the lack of project flexibility. Then, the solution often led to disappointment and bad feedback.

The *data-driven user experience (UX) design (3DUX) process* avoids this problem, like all human-centered development processes, by assuming from the beginning that an iterative process is necessary where new requirements are discovered in testing and during design. Further, this approach assumes that iterations occur before technical development (see Chapter 5, Section 5.2).

In this chapter, we'll introduce you again to a small iterative procedure to help you sharpen requirements piece by piece and move from simple content requirements that are still formulated quite generally to very concrete requirements that are easy to check.

In this context, however, we must issue a clear warning: Do not formulate detailed requirements before you know what is needed! If you have worked cleanly in the last stage (when analyzing the context of use), this should not be a problem. You should have received plenty of input through observation and interviewing. If not, now is the time to take another step back and collect actual data. Basing requirements on assumptions and formulating them as if you already know what is needed is a direct path into expensive and probably unnecessary feature development. Let's start at the very beginning—with the results of the context of use analysis.

10.1 Consider Your Personas or Empathy Maps

As a result of your context analysis, you should have developed a good feeling for your target group at this point. In the best case, you have already created empathy maps or personas (see Chapter 9, Section 9.4.3). Now is the time to pull out those representations and look at them in detail. We'll focus on the empathy map in this section. Empathy maps have the advantage of being a predetermined structure, making it easier for us to guide you through each area. However, if you have developed personas, you can proceed in the same or similar way since you have probably compiled similar information.

In the first section of your empathy map, you looked at the *thoughts and emotions of your target audience*. What does your target audience believe? What are they worried about? All of these aspects are now looked at again in more detail and checked to see whether you're already in a position to derive general requirements for your solution from them.

Imagine, for example, that you were designing software for telephone counseling staff. A possible empathy map would perhaps show that the people who work in the telephone counseling service can be characterized in the following ways:

- They are proud of their work and feel they are making a significant contribution.
- They are stressed for fear of being unable to give the right advice in every situation or not having enough time for all callers.
- They would like to help as many people as possible.
- They have guidelines on how long phone calls to individual callers should last.

Which of these statements can you use now for your later requirements? Since it seems essential to how many people can be helped and whether the conversation was helpful for the callers, you could include these two points to give feedback to the employees. Therefore, initial points you could pick out for further design could be described in general terms in the following ways:

- It must be evident how many calls have already been made that day.
- It must be evident how well the calls went from the callers' point of view.
- It must be apparent how long a call will take.

Now, we are still quite general. The three points do not follow a predefined structure with which we record the requirements. For our example, we are concerned only with collecting. You can then proceed in the same way with the other areas. You may end up with a relatively long list of points to consider. But how do you document these points now to derive a suitable solution?

In the first general step, an important point is to remain *solution-neutral* in any case! After all, we don't know yet what kind of solution we'll develop. Especially when you document the requirements for your system, product, or service for the first time, you won't know many things as well as you will in one or two iterations. Thus, your solution space may change entirely once again. To enable this change, formulate your requirements in a solution-neutral way. What does that mean? Let's illustrate this concept with an example.

Think again about the software for telephone counseling you're currently designing. The first general requirement we formulated informally was "It must be evident how many calls have already been made that day." That statement is solution neutral for now. When you write, "A display at the bottom right of the screen must count up the number of accepted calls in increments of one," very little room for implementation is left. Perhaps it would be better to display the answered calls not as an absolute number, but as a relative one, for example, by showing 9 out of 10 incoming calls answered. Alternatively, a visual display could be dispensed with altogether, and instead, an audible indicator could announce the number after the call has ended. Or, or, or. Any of these proposed implementations would satisfy our general requirement. The other way around, however, would not.

Now that we know we should be solution neutral, at least at the beginning, let's take it further and ask, "How do I formulate requirements now?"

10.2 Use Cases and User Stories: How Not to Do It and How to Do It Anyway

In this section, we would like to show you two variants of how requirements are usually documented from a technical point of view. In a second step, we'll explain why we don't think this technical focus is a particularly good idea and show you a better way.

10.2.1 Use Cases

One of the most common ways to document requirements is the creation of use cases. *Use cases* are scenarios that can occur when users operate a system, product, or service. Typically, they describe the behavior of an existing solution or a fictitious solution that is visible to the users.

Use cases are typically represented with use case diagrams. *Use case diagrams* graphically represent how users interact with a system (e.g., software), where a use case diagram often represents multiple and different use cases that a user may experience with a system, as shown in Figure 10.1.

Use case diagrams initially describe the interaction between an actor and a system. In the case of technically complex systems, actions are also listed in such a diagram in practice that are not executed by a human being but executed by another (technical) system. However, if your goal is to optimize usability and UX, these actions are only relevant to a limited extent. However, they make the diagram more complicated and should be avoided from the point of view of the usability and UX expert.

Further criticism of use case diagrams comes from Joe Natoli (2017): For us to ensure that the implementation accomplishes what the users are trying to do—that is, solves their job—an essential task is to understand the user's goals. While occasionally use case diagrams include user goals, in most cases, they simply depict sequences of tasks, as shown again in Figure 10.1.

This limitation is also related to the fact that choosing a good design only based on such a sequence of tasks becomes difficult if the value of this one requirement (or in the case of use case diagrams, what this task brings to users) is unclear. Designers must then fish around in the mud and hope they hit on a practical design. This task becomes all the more difficult the less the people designing are integrated into the context of use analysis and the more distant the target group is from their world of experience.

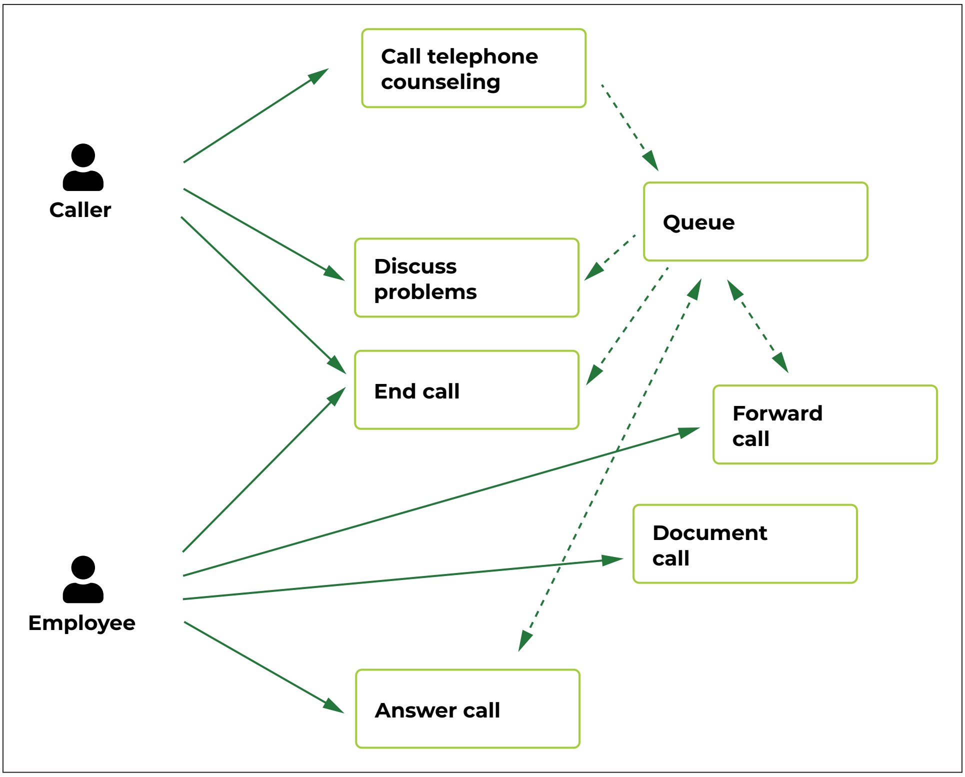

Call telephone counseling
Caller
Queue
Discuss problems
End call
Forward call
Document call
Employee
Answer call

Figure 10.1 Use case diagram for our telephone counseling service example. Representation as a diagram can quickly become confusing.

10.2.2 User Stories

The second frequently used format is that of user stories. *User stories* capture requirements for the system, the product, or the service, formulated from the user's point of view. This formulation follows a simple format and always has the same structure. A widely used format (several similar ones are available) for a user story is shown in Figure 10.2.

As a **USER** I want **FUNCTION**, so that **ACTION**.

Figure 10.2 Formula of a user story (based on Cohn, 2004).

This recipe means you can easily and quickly understand which target groups are the intended targets of a particular function within a system, product, or service. This information is quite handy when you have different target groups, and a weighting or a sprint is to be used primarily for the needs of one target group. Also, the recipe allows the design team to consider that group's knowledge and experience.

Unfortunately, user stories—like use cases—often "stop" at the task level. This limitation is not due to the format of user stories per se but to the way user stories are used. User stories directly ask *what* the target group can do with a specific function. This formation addresses user goals only indirectly, and we end up with the same problems as with use case diagrams: It becomes difficult to understand what *added value* a particular function offers from the user's point of view.

Once again, you risk including tasks in user stories because specific goals are now achieved through certain tasks, not because it is the best way to do so. We want to suggest that, at least at the beginning of development, you approach the development of your requirements in a broader, more general, and a bit more idealistic way.

For this approach, however, you must know the context of use of your users very well. More precisely, you need to know the actual underlying goals of your users to integrate these goals into your user stories. If this is the case and you focus on the goal, you at least have the chance to think much more innovatively than may currently be the case.

To achieve this task, Joe Natoli offers a slightly modified version of user stories that addresses this shortcoming of task fixation. His form of user stories picks up directly on the users' goals and names them as such. Other authors have also revised the original format and added goals (e.g., Marcano, 2011). However, we'll stick here to Natoli's improved formation, which will be further supplemented in the next section. The new format that Natoli proposes follows the pattern shown in Figure 10.3.

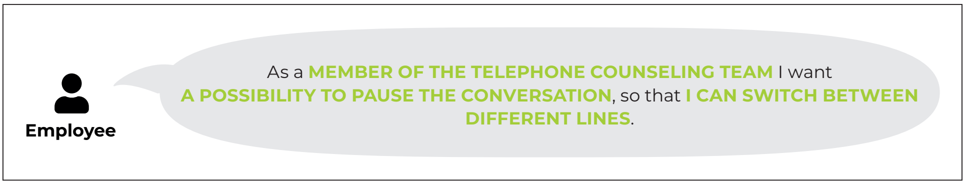
The **USER** wants **FUNCTION**, to achieve the goal of **USER GOAL**.

Figure 10.3 Improved format of a user story (based on Natoli, 2017).

Formulating such user stories gives you a decisive advantage over conventional formulation: You remain focused on the (actual) user goals. You can then more easily find a suitable solution later when developing design solutions instead of just picking any solution. Let's illustrate this difference with an example.

Think again of our employees in the telephone counseling service. A typical user story, formulated in the "conventional" way, is shown in Figure 10.4.

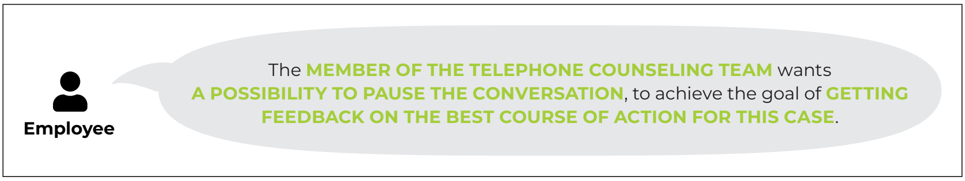
This user story describes the task well in itself. As long as we assume that the employees will continue to work with standard telephone systems in the future, the wording will probably not have any significant adverse effects. However, due to the mentioned line change, we can already determine that the task is a change in the call partner.



As a **MEMBER OF THE TELEPHONE COUNSELING TEAM** I want **A POSSIBILITY TO PAUSE THE CONVERSATION**, so that **I CAN SWITCH BETWEEN DIFFERENT LINES**.

Figure 10.4 Conventionally formulated user story.

In our example, we assume that we have already created an empathy map for the employees in the telephone counseling service. Looking at the thoughts of the target group summarized there tells us that uncertainty about doing the right thing is a significant driver. Thus, behind the hold function, in this case, is not the desire to hold someone and talk to someone else but the desire for security. Thus, a reformulated user story could look like the one shown in Figure 10.5.



The **MEMBER OF THE TELEPHONE COUNSELING TEAM** wants **A POSSIBILITY TO PAUSE THE CONVERSATION**, to achieve the goal of **GETTING FEEDBACK ON THE BEST COURSE OF ACTION FOR THIS CASE**.

Figure 10.5 Reformulation of the user story.

If you were to describe the formulation using a rule, the rule would be “Describe what success means to the user, not what they need to do.” By formulating a user story in this way, you accomplish two things at once:

- First, you ensure that users achieve their goals no matter what you design. They are not just performing some task; they are doing something that gets them closer to their original goal. That’s good!
- Second, you suddenly have a much larger solution space. Maybe a supervisor in the room can provide support. Maybe a case database or a chatbot is available, maybe just a colleague on another line. But you don’t have to define those resources at this point.

This type of user story is advantageous if you’re still at the beginning and need an overview of the product. In other words, you still have different approaches and have not decided what it should be and how exactly you want to design your product. The more time you spend developing the product, the more precise the picture of the product will become for you, so at some point, you’ll need a different set of requirements. More on that topic in a moment. Let’s introduce another extension to user stories from Joe Natoli.

User Stories Extension: Part 1

In addition to the core sentence of a user story, Natoli also brings two other sentences into the description. These sentences do not aim to optimize the product. The sole purpose of the two additional sentences is to help you, and everyone else involved in the process, prioritize user stories. You have probably used the sentences shown in Figure 10.6 or a similar formulation already.

If you ask your development team which requirements are essential, then all requirements are always essential. For every requirement, usually at least one user story about a customer who needs the requirement should be clear: Without this requirement, the customer definitely drops off, and everything explodes.

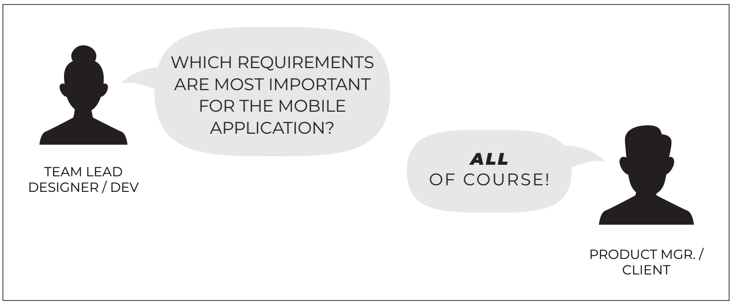

WHICH REQUIREMENTS ARE MOST IMPORTANT FOR THE MOBILE APPLICATION?
TEAM LEAD DESIGNER / DEV
ALL OF COURSE!
PRODUCT MGR. / CLIENT

Figure 10.6 A situation we have often experienced: Seemingly, everything is equally important.

We don't think that's true (the explosion and the equally essential requirements)—at least not always and everywhere. Thus, we want to provide you with a tool to help prioritize requirements, which is where the two extension sentences to user stories from Joe Natoli come back into play. The first set is shown in Figure 10.7.

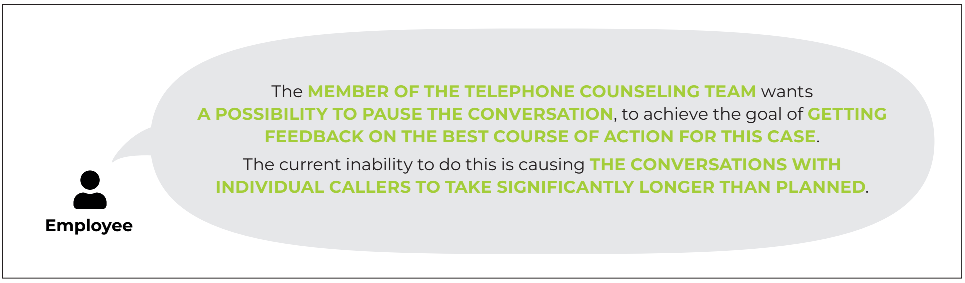
The current inability to do this is causing an **ADVERSE EFFECT FOR THE ORGANIZATION**.

Figure 10.7 First extension of a user story (based on Natoli, 2017).

In this way, Natoli addresses the problem of requirements being brough into a development project from different directions, but the user is rarely the focus of these requirements. Combining the user ("What can't they do right now?") and the resulting disadvantages for one's organization make it much easier to weigh.

Again, let's consider our telephone counseling example. Our new user story is shown in Figure 10.8.

If the number of calls answered or the average call duration was an essential criterion for the telephone counseling service, it would be clear now that providing a facility for internal counseling should be a high priority. On the other hand, if only caller satisfaction were important, the telephone counseling team would perhaps decide that the function need not necessarily be implemented now or at all.



The **MEMBER OF THE TELEPHONE COUNSELING TEAM** wants **A POSSIBILITY TO PAUSE THE CONVERSATION**, to achieve the goal of **GETTING FEEDBACK ON THE BEST COURSE OF ACTION FOR THIS CASE**.

The current inability to do this is causing **THE CONVERSATIONS WITH INDIVIDUAL CALLERS TO TAKE SIGNIFICANTLY LONGER THAN PLANNED**.

Figure 10.8 Example of the user story with the first extension.

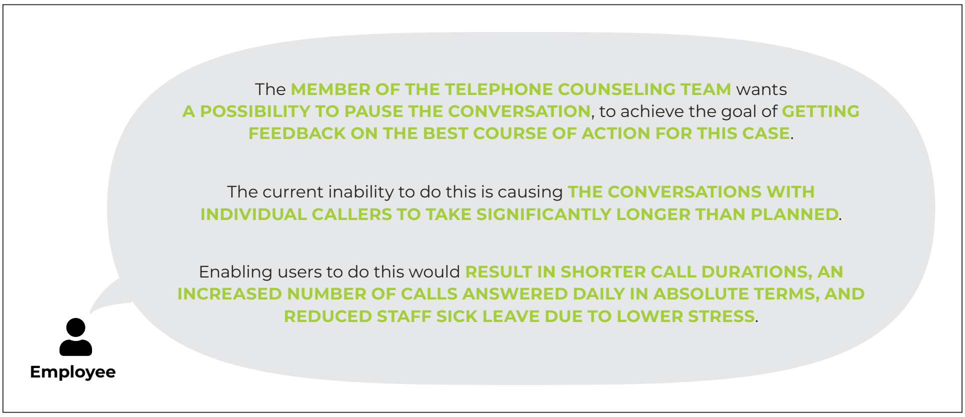
User Stories Extension: Part 2

As further support for prioritization, a second extension of the original user stories concept is available. Just now, we talked about what disadvantages arise from the current situation. Now, we'll look again explicitly at the future benefits, as shown in Figure 10.9.

Enabling users to do this would **ADD MEASURABLE VALUE TO THE ORGANIZATION**.

Figure 10.9 Second extension of a user story (based on Natoli, 2017).

The task now is to explicitly show what the organization gains from implementing this desired function. Transferred to our example, including the first extension, this results in the following user story, as shown in Figure 10.10.



The **MEMBER OF THE TELEPHONE COUNSELING TEAM** wants **A POSSIBILITY TO PAUSE THE CONVERSATION**, to achieve the goal of **GETTING FEEDBACK ON THE BEST COURSE OF ACTION FOR THIS CASE**.

The current inability to do this is causing **THE CONVERSATIONS WITH INDIVIDUAL CALLERS TO TAKE SIGNIFICANTLY LONGER THAN PLANNED**.

Enabling users to do this would **RESULT IN SHORTER CALL DURATIONS, AN INCREASED NUMBER OF CALLS ANSWERED DAILY IN ABSOLUTE TERMS, AND REDUCED STAFF SICK LEAVE DUE TO LOWER STRESS**.

Figure 10.10 Example of the user story with the first and the second extension.

Now, you have not only disclosed the disadvantages of the current situation on one side but also made the possible advantages of an adaptation transparent. We find this exercise helpful because, in many developments, the cost of implementing a feature is often discussed at length. However, we rarely experience the case where the costs are named if everything remains as is (i.e., the *opportunity costs*). These costs remain hidden in most cases because they are "incurred anyway."

We encourage you to make these costs visible, which is the only way to make informed decisions. Not every feature needs to be implemented. It's just that a fair chance was given for every feature.

But what if you realize, when formulating your user stories, that you can neither name disadvantages nor formulate clear advantages? To put it bluntly, "Then maybe it just wasn't that important...." If you have requirements without clear advantages or clear disadvantages, you should urgently consider removing them from your list or from your backlog. Advantages or disadvantages do not always have to be external. Examples of internal dimensions can also include better communication, higher satisfaction, regulatory compliance, or some internal goal. Only if no goal exists and no internal or external benefit could be determined should you be consistent in not implementing a requirement.

Should I Really Make the Effort?

If you don't know your target group well yet, you look for opportunities to get to know them better. If you know your target audience well and have worked closely with them for many years, you may just be feeling, "What's the point?" After all, it's all so clear to you. Yes, that may be so. But we want to caution once again about the paradox that you, as a person involved in product development, know too much and too little at the same time compared to your target group:

- *Too much* because you know and consider how it works, the internal restrictions, the economic goals, and the technical and regulatory framework in development.
- *Too little* because you're not in the daily routine of your target group and, therefore, cannot know everything about their processes and general conditions.

This disconnect is always the case and is not always a bad thing. The important thing is that you manage to convey the knowledge about the goals and background in a way that gives everyone involved in the development access to the knowledge, which is where extended user stories can help. But what do you do now that you have defined your user stories and want to implement them? Everything is clear and "just needs to be done now." Then, another type of requirement can help, which we'll explore next.

10.3 From User Stories to Requirements

Outside the software industry, user stories are less common. Instead, a common tactic is to work with "classic" requirements. Classic requirements are commonplace, especially in areas where requirements must be tracked and compliance must be proven. But what does working with classic requirements mean? In this case, what the system, product, or service must be able to do later is usually recorded in text form. In engineering theory, additional distinctions are made between *fixed*, *range*, *minimum*, and *maximum requirements*.

10.3.1 Fixed Requirements

If you formulate a fixed requirement, then exactly one correct answer exist. For example, you could define the color of a specific warning symbol as fixed because a standard prescribes a value. Then, your requirement would contain this defined color. A typical warning triangle (with a red that is often defined as #ff0000) is shown in Figure 10.11.

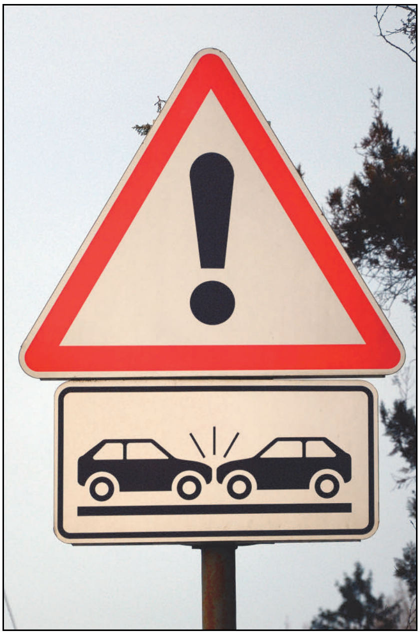


Figure 10.11 Example of a fixed requirement: the color of a warning triangle.

Thus, any other color for your warning triangle would be incorrect. You just can't make your warning triangle #7F1412, even though most people would say, "That's red!" You can see that red is not the same, as shown in Figure 10.12.

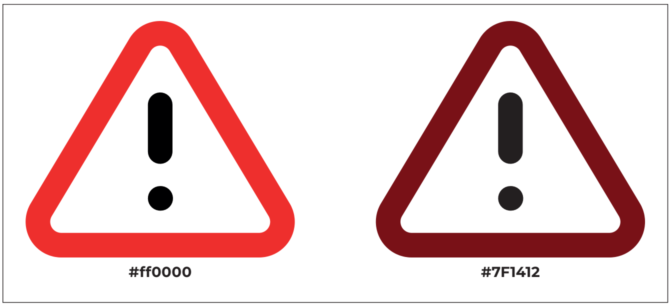


Figure 10.12 Comparison of two warning triangles: one in #ff0000 and one in #7F1412.

By defining a fixed requirement, you usually achieve a precise result for all developers. In addition, you can usually easily decide whether the requirement has been implemented correctly or not.

From our point of view, fixed requirements are helpful when you want to ensure that something is implemented the exact way and not differently. Warning symbols are an excellent example because they are often subject to regulatory requirements that you must comply with.

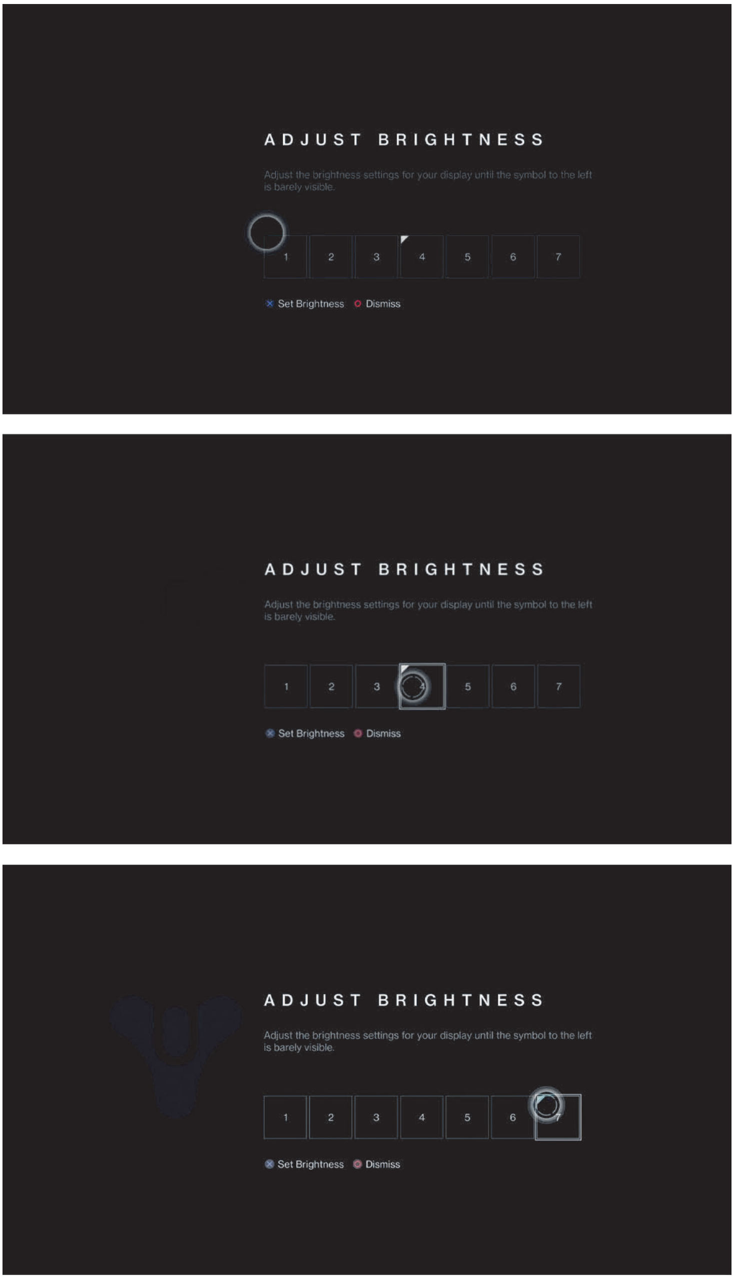
Fixed requirements can also be helpful if you want a consistent style or look and feel in your solution. However, fixed requirements in the user interface (UI) area are usually not formulated as text but instead provided as a digital or interactive style guide. We'll cover using the *style guide* as a tool in more detail in Chapter 11, Section 11.12.8. However, you can already remember that the style guide is nothing more than a requirements catalog for the UI.

10.3.2 Range Requirements

In contrast to fixed requirements, range requirements do not specify a single value that must be achieved. Instead, as the name implies, they specify a range in which the measured value must later fall. In engineering, range requirements occur frequently.

An example would be the adjustment range for a height-adjustable table, which should range between 25 and 50 inches. The brightness of a monitor is also usually specified between a lower limit and an upper limit. In everyday life, we are also familiar with the range of requirements for car tires, for example, which can be inflated between a minimum pressure and a maximum pressure. Another example is the "required" room temperature in an office which usually ranges between 68 and 72 °F.

Range requirements also exist in usability and UX design. One example might be a button that should be at least large enough to be safely operated with a finger but small enough so you can fit five more buttons on the casing. Another example is often found in computer games, for instance, when the player can adjust the game's brightness within certain limits, as shown in Figure 10.13.



ADJUST BRIGHTNESS

Adjust the brightness settings for your display until the symbol to the left is barely visible.

ADJUST BRIGHTNESS

Adjust the brightness settings for your display until the symbol to the left is barely visible.

ADJUST BRIGHTNESS

Adjust the brightness settings for your display until the symbol to the left is barely visible.

Figure 10.13 Various brightness settings in Destiny 2. The value should be set so that the icon on the left is barely visible.

In all the examples we just presented, we have specified a lower limit and an upper limit in each case. We're really working here with a subclass of range requirements, called *interval requirements*. Range requirements also include the minimum and maximum requirements, which we'll examine in more detail in the following two sections.

Minimum Requirements

Minimum requirements do what the name suggests: They define what must be achieved as a minimum. You're probably familiar enough with minimum requirements in your private life in various areas, such as the following examples:

- To ride some roller coaster, the sign says you must be at least 55 inches tall and 16 years old.
- To open a free checking account with a certain bank, you must provide a minimum monthly cash inflow of $750.
- To buy a house on credit, you must have at least $1,500 of free capital each month.

However, minimum requirements are also relatively common regarding usability and UX. For example, minimum distances between elements are often defined as minimum sizes for elements and fonts to ensure that they are easy to use and read. A typical minimum requirement would be that all buttons must have a minimum size of 0.4 × 0.4 inches or that the distance between two buttons is at least 0.08 inches. Similarly, you could specify that you want to maintain a minimum contrast between two adjacent color areas or a minimum distance from the edge of the screen. All these would be minimum requirements, as shown in Figure 10.14.

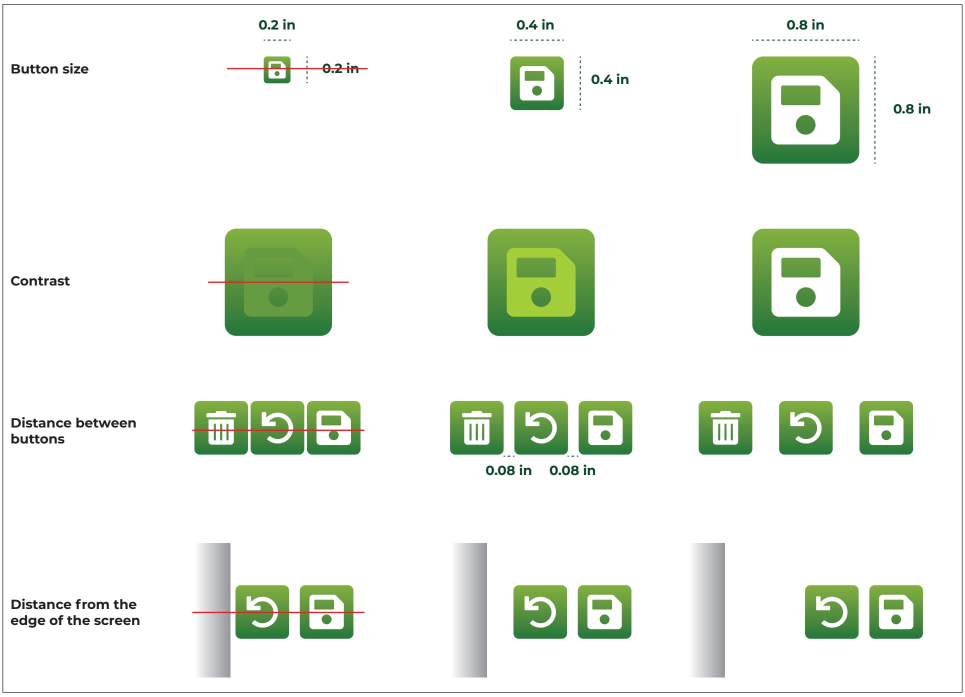


Figure 10.14 Example minimum requirements for the size of buttons, minimum contrast values, spacing between elements, and the distance from elements to the edge.

In the case of a minimum requirement, you always check whether the minimum level has been reached. The actual value is unimportant if above the minimum level. Thus, if you define a button size with a minimum requirement of at least 0.4 × 0.4 inches, they could just as well be 40 × 40 inches. Thus, considering in advance if no maximum exists make sense.

Maximum Requirements

Maximum requirements are the opposite of minimum requirements. These requirements do not specify a hurdle you must jump over to "get in," but they specify the limit below which you must stay (see Figure 10.15). A few examples from the private sphere include the following requirements:

- In a parking garage, the maximum allowed vehicle height is 74 inches.
- No more than 0.01 oz of free chlorine may be found in the drinking water.

Even with sports equipment, you often need to deal with maximum requirements. For example, a trampoline might be limited to a maximum of one person and may not be used if that person weighs more than 330 pounds. A typical maximum requirement is a price. People often specify how much a component or product may cost as a maximum.

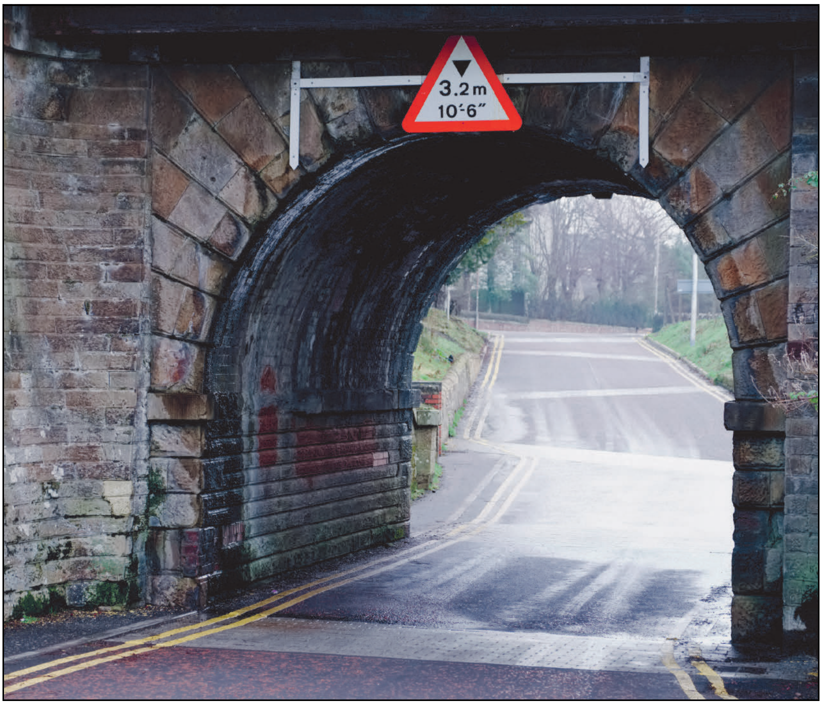


Figure 10.15 Example of a maximum requirement.

As you can see, maximum requirements are everywhere in your everyday life.

Maximum requirements also exist in the area of usability and UX. These requirements often relate to the number of elements, such as the following:

- A security code must not contain more than 6 digits.
- No more than 60 seconds may elapse before the security code is entered.
- The maximum idle time is 5 minutes after logging in before logging out automatically.

Maximum demands are often set when something bad happens as soon as you exceed the value. More chlorine in drinking water? Unhealthy! Taller car? Roof broken! Heavier than 330 pounds? Risk of injury! So set maximum demands when otherwise something bad happens, or you have a defined limit "upwards."

10.3.3 Requirement versus Wish

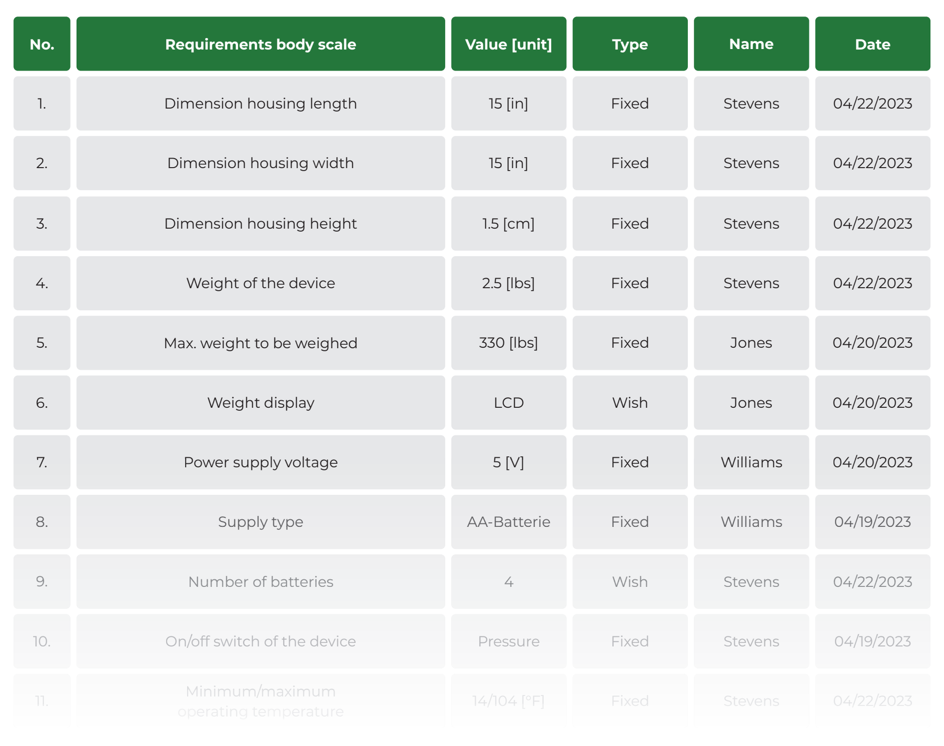
Besides how a requirement can be structured in terms of content, an important question is recognizing which requirement is essential. Unlike our user stories, unfortunately, no context is available to help with this problem. Mostly, this issue is solved simply by marking the requirements as either a requirement or a *wish*. Wishes are "nice to have, if there is still time." But let's be honest: When does that happen during development?

10.3.4 Documenting and Formulating Requirements

Whether, what, and how much you document depends on your goal. If you're a lone wolf bringing your first app to market, then a detailed requirements document, as shown in Figure 10.16, might be too much. Suppose you're developing a medical device that will later be approved in European and US markets. Your requirements list should be as complete as possible and traceable to the *user's needs* and the later implementation.

So how much documentation you need depends on what you want to achieve. In general, we advise creating documentation that is as lightweight as possible (while still, of course, complying with any legal requirements that may apply). The goal of formulating requirements is not a perfectly crafted document but a perfectly crafted product.

We've had good experience documenting requirements for a user's workflow with a system and documenting requirements for the UI itself graphically. This step is usually done for the interface itself as a *style guide* (see Chapter 11, Section 11.12.8) and for the workflows in the form of *screen flows* (see Chapter 11, Section 11.8).



No.	Requirements body scale	Value [unit]	Type	Name	Date
1.	Dimension housing length	15 [in]	Fixed	Stevens	04/22/2023
2.	Dimension housing width	15 [in]	Fixed	Stevens	04/22/2023
3.	Dimension housing height	1.5 [cm]	Fixed	Stevens	04/22/2023
4.	Weight of the device	2.5 [lbs]	Fixed	Stevens	04/22/2023
5.	Max. weight to be weighed	330 [lbs]	Fixed	Jones	04/20/2023
6.	Weight display	LCD	Wish	Jones	04/20/2023
7.	Power supply voltage	5 [V]	Fixed	Williams	04/20/2023
8.	Supply type	AA-Batterie	Fixed	Williams	04/19/2023
9.	Number of batteries	4	Wish	Stevens	04/22/2023
10.	On/off switch of the device	Pressure	Fixed	Stevens	04/19/2023
11.	Minimum/maximum operating temperature	14/104 [°F]	Fixed	Stevens	04/22/2023

Figure 10.16 A typical requirements list. For some developments, something like this is just right; for others, this is too much effort.

Any technical requirements that otherwise exist for the product or that have to do with the interaction of the product and the user, but are not directly visible on the UI (e.g., “After three unsuccessful login attempts, the user account will be locked”) should also be recorded as requirements. However, to work as work instructions, all requirements must be written as clearly as possible to be meaningfully implemented by the subsequent implementation team, even if the team members don’t understand much about the situation or processes. Thus, you should make the requirements as concrete as possible and provide them with values rather than descriptions.

For example, a reasonable requirement would name the shade of red #ff0000, while a lousy requirement would just say “in red.” An even worse requirement would read, “In a color that catches the viewer’s attention.”

Can you see the difference? The first version would ensure that the correct color is used. With the second version, three people would probably use three shades of red. The results might vary entirely from the last version. Perhaps a red would be chosen by chance, but perhaps a yellow, orange, or poison green, as shown in Figure 10.17. If you don’t care about the result: leave it open. If you want uniformity, define concrete values.

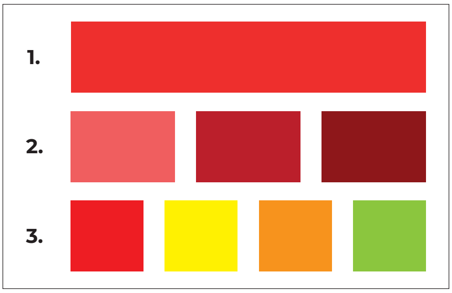


Figure 10.17 Colors that can result from the three different requirements: "in red tone #ff0000" (1), "in red" (2), and "in a color that attracts the viewer's attention" (3).

Concrete values also have an advantage in making the *requirements measurable* and their *fulfillment easy to assess*. Concrete values make it easy to prove later that a requirement has been met. If you require the red tone to be #ff0000, you can technically verify the correct value is used. If you require that the hue be attention-grabbing, then you have to force-test that, even though that may have only been a minor requirement. Thus, a requirement is not just how something should be, but whether you need to prove it.

Even though we are absolute fans of testing: You should only ever test when a good reason exists to do so. This rule is especially true for the medical technology sector. In this industry, you must prove that your requirements have been met. In our opinion, however, you should focus your testing power on showing that the process fits, that there is understanding, and that the operation works safely. However, if you have to test *everything* due to regulatory requirements, it is easy for the critical content in the test to receive too little attention.

For you to proceed in this way, you must ensure that the required values you define in advance are the right ones, which brings us to the next topic.

10.3.5 Where Do the Values in the Requirements Come From?

Turning a rough idea into a concrete, measurable value is one of the most challenging tasks. At this point, we can't give you a handbook with the perfect values for everything. The tasks are too varied, and the requirements are too complex. Instead, we would provide some suggestions that you can use in the future to derive values when you need them.

In Chapter 7, Section 7.2, you learned quite a bit about the perceptual capabilities of humans. You should remember this knowledge as a basis for all human perception requirements.

In addition, you'll find many different national, International Electrotechnical Commission (IEC), or International Organization for Standardization (ISO) standards that will help you define suitable limit values for different circumstances.

We usually derive values from users and their capabilities when we look for values. How big is the typical finger? So how big does a touch surface need to be? What contrast ratio can a person recognize well?

If you're in "normal" software development, you also don't need to derive every value exactly, but often, it is sufficient if you have defined it cleanly. If you work in medical technology, you cannot avoid the clean derivation of requirements and must prove your assumptions. Under certain circumstances, this need can go so far that you must perform and document your measurements and tests to formulate your requirements accurately. In most cases, however, showing your work will be completely unimportant.

Formulating Requirements Means Working with Many Disciplines

A note about the required disciplines: As an expert in usability and UX, you can only formulate some requirements independently. For many other areas, you may need additional expertise—from the area of technology, the application, and perhaps the area of data protection or law. This additional expertise may even be the case if the project is "only" about the design of the UI.

For example, we would like to mention "double opt-in." In this case, the General Data Protection Regulation (GDPR) in the EU and current case law force you to design your UI or your interaction with users in such a way that they permit you twice to contact them, once on the website, for example, and then again via a link in an e-mail. This requirement in data protection must be directly reflected in your interaction.

The same can happen with IT security or other areas. Therefore, get all the expertise on board quickly to ensure the requirements fit.

If you work in a *regulated field*, you may even find that usability or UX requirements that do not arise directly from the context of use analysis but are already inscribed in norms and standards. Considering these norms as early as possible in the design is essential. Usually, norms or standards can be found whenever a general potential for harm exists.

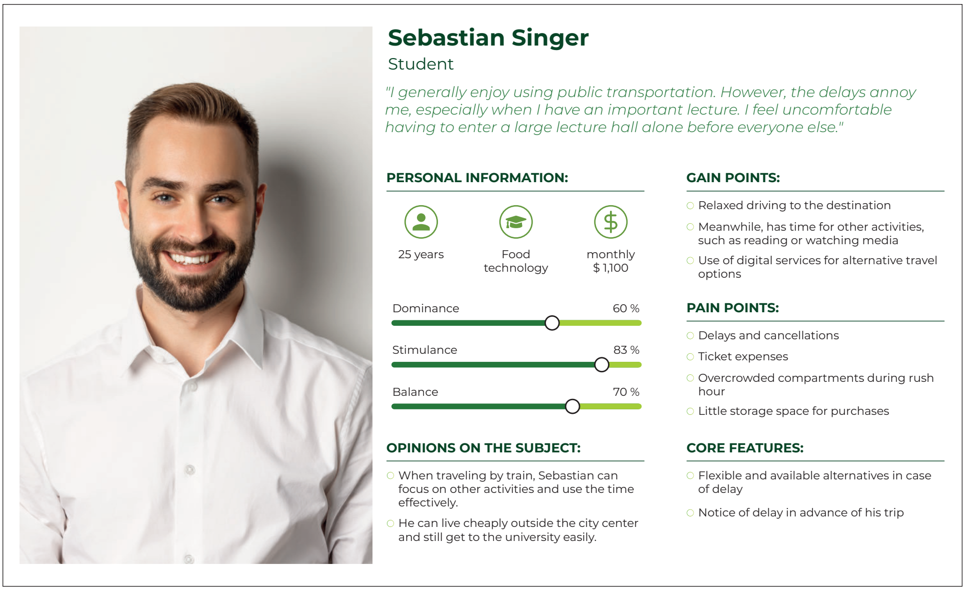
Thus, for warning symbols or alarms, you're much more likely to find an already-defined requirement than for a logon. In general, however, our experience shows that norms and standards now exist for almost every area. The most challenging part is sometimes to identify the appropriate standard because the names of standards do not necessarily match your language.

In the next section, we'll consider how requirements might be formulated for the projects of our three sample companies.

10.4 Formulating Requirements for moveHUB

When we last saw our founders, back in Chapter 9, Section 9.5.1, they were deep in the context of use analysis. Based on they've data obtained, they identified the two personas: *Sebastian "The Student" Singer* and *Sina "The Student" Scott*. The three then collected other data to ensure the personas were correct.

moveHUB has derived the B2C mobility platform requirements based on these personas. For this task, they first looked at the persona *Sebastian "The Student" Singer*, as shown in Figure 10.18, and are now formulating user stories together. They decided to use the detailed format with all extensions.



Sebastian Singer

Student

"I generally enjoy using public transportation. However, the delays annoy me, especially when I have an important lecture. I feel uncomfortable having to enter a large lecture hall alone before everyone else."

PERSONAL INFORMATION:

25 years

Food technology

monthly $ 1,100

Dominance 60 %

Stimulance 83 %

Balance 70 %

OPINIONS ON THE SUBJECT:

- When traveling by train, Sebastian can focus on other activities and use the time effectively.
- He can live cheaply outside the city center and still get to the university easily.

GAIN POINTS:

- Relaxed driving to the destination
- Meanwhile, has time for other activities, such as reading or watching media
- Use of digital services for alternative travel options

PAIN POINTS:

- Delays and cancellations
- Ticket expenses
- Overcrowded compartments during rush hour
- Little storage space for purchases

CORE FEATURES:

- Flexible and available alternatives in case of delay
- Notice of delay in advance of his trip

Figure 10.18 Sebastian "The Student" Singer persona for moveHUB.

Note

Full PDF versions of Figure 10.18 through Figure 10.20 are available for download at *www.rheinwerk-computing.com/5778* under the product supplements.

We'll pick out one of the user stories at this point:

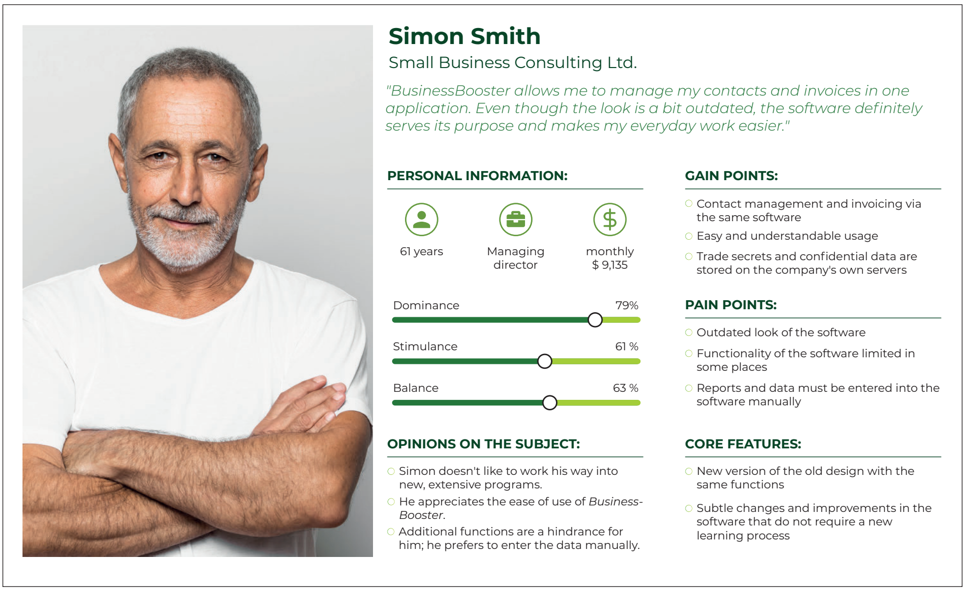
> *As Sebastian "The Student" Singer, I want to arrive at the lecture hall on time to not sit alone in a large lecture hall or be late in front of everyone.*
>
> *The current inability to do this means that I sometimes have to fritter away time (rarely), but I usually miss a lecture because I'm embarrassed to be late in front of everyone.*
>
> *Enabling Sebastian to do this would allow him to hear the complete study material in lectures, reducing his stress significantly.*

The three do the same with all the other challenges identified by the target group. All in all, they have a list of over 40 user stories that need to be tackled. Although the three founders are aware that they are facing a massive amount of work, they are optimistic: By formulating these user stories, they have become aware that they can now more easily decide whether they need a feature and, if so, how it should be implemented.

10.5 Formulating Requirements for BusinessBooster

Markus Meier has already taken a big step forward with his company and his customer relationship management (CRM) business intelligence (BI) tool by getting involved in a new world. User-centeredness was previously a foreign word to him, and putting technology on the back burner for the time being is not easy for him. It is, therefore, not surprising that he initially wants to stick to his tried-and-tested principle of the requirements list when drawing up the requirements.

Most recently, back in Chapter 9, Section 9.5.2, the team defined the personas *Simon "Small Business" Smith*, *Janine "Young Digital" Jones*, and *George "Growth" Giant* based on available data and intensive discussions. For Markus, the persona Simon "Small Business" Smith, shown in Figure 10.19, is the most important one, so the team tackles this persona first.



Simon Smith

Small Business Consulting Ltd.

"BusinessBooster allows me to manage my contacts and invoices in one application. Even though the look is a bit outdated, the software definitely serves its purpose and makes my everyday work easier."

PERSONAL INFORMATION:

61 years

Managing director

monthly $ 9,135

Dominance 79%

Stimulance 61 %

Balance 63 %

GAIN POINTS:

- Contact management and invoicing via the same software
- Easy and understandable usage
- Trade secrets and confidential data are stored on the company's own servers

PAIN POINTS:

- Outdated look of the software
- Functionality of the software limited in some places
- Reports and data must be entered into the software manually

OPINIONS ON THE SUBJECT:

- Simon doesn't like to work his way into new, extensive programs.
- He appreciates the ease of use of *BusinessBooster*.
- Additional functions are a hindrance for him; he prefers to enter the data manually.

CORE FEATURES:

- New version of the old design with the same functions
- Subtle changes and improvements in the software that do not require a new learning process

Figure 10.19 Simon "Small Business" Smith persona for BusinessBooster.

The most critical and frequent feedback from the Simon "Small Business" Smith persona concerns the lack of automation capabilities for reports. The team, therefore, decides to define the following raw requirements:

- Each data module must provide the data it contains via an interface within the software.
- A new "Analytics" module will tap into data from all other modules.
- The "Analytics" module must also have options for evaluating the data.

Especially with the last point, the team quickly gets into the details after formulating the general requirement. Requirements such as "The display is done as a pivot table" and the "Pivot table must be able to display at least 10,000 rows" arise.

That's when the external consultant from the hired data-driven UX design agency raises his hand. He makes it clear that, based on the current user data, we still don't know what data Simon "Small Business" Smith needs and how it should be displayed. Nor does he accept the comments from the team that a pivot table is the only sensible solution. At first, Markus's team was shocked by this interruption. Soon after, however, the team realized that the external consultant was right: The team got lost in the technical implementation again and did not focus on the user or on knowledge about the user.

The team decides to jump back into the context of use analysis and dig deeper into what reports they want: What exactly do we want the reports to show? Should they be ready-made reports? Should you be able to customize them to your own needs? What would be an ideal report?

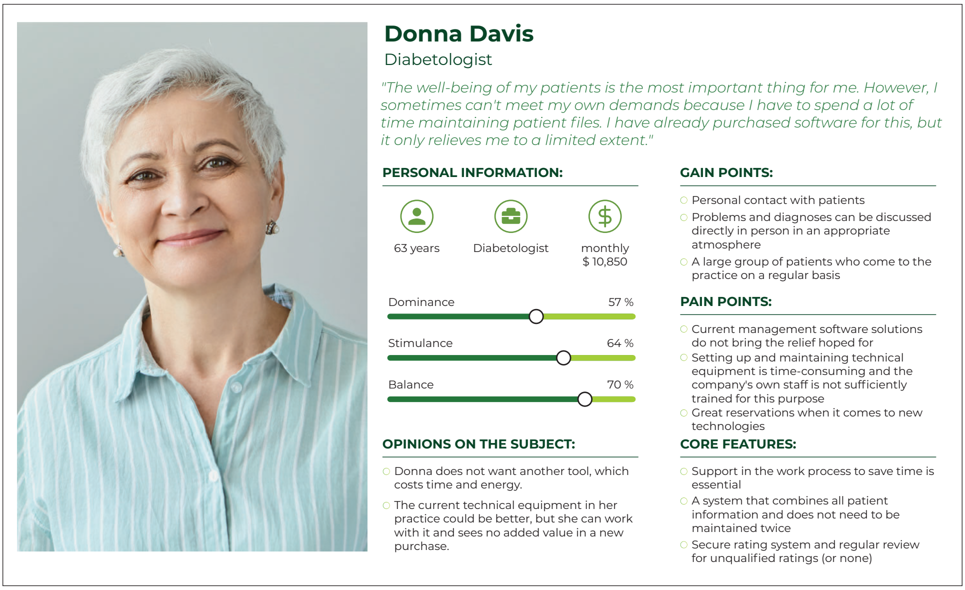
Together with the external consultant, the team conducts in-depth interviews with customers and potential customers who match the persona of Simon "Small Business" Smith. To be on the safe side, they also take a few representatives of the personas Janine "Young Digital" Jones and George "Growth" Giant with them. In summary, the interview results show that pivot tables are indeed an excellent representation option. However, 10,000 rows are far too few for the desired reports. In addition, graphical evaluations should also be possible. Simple pie charts, in particular, are high on customer wish lists.

With these new insights, Markus's team jumps back into the requirements definition and completes the list accordingly. Since this succeeds sufficiently well, the team decides to produce the first designs next.

10.6 Formulating Requirements for Healthyfy

When we last met with the Healthyfy team, back in Chapter 9, Section 9.5.3, they had just found out, via focus groups, that the *Donna "The Diabetologist" Davis* persona, as shown in Figure 10.20, would not be satisfied with the application in its current design.

Donna was mainly concerned about the technology and her duty of care. Since her practice is consistently full anyway, she doesn't see any advantage in getting new patients through Healthyfy. In fact, she is somewhat put off by the patient reviews. Donna's primary need is time savings.



Donna Davis

Diabetologist

"The well-being of my patients is the most important thing for me. However, I sometimes can't meet my own demands because I have to spend a lot of time maintaining patient files. I have already purchased software for this, but it only relieves me to a limited extent."

PERSONAL INFORMATION:

63 years

Diabetologist

monthly
$ 10,850

Dominance 57 %

Stimulance 64 %

Balance 70 %

OPINIONS ON THE SUBJECT:

- Donna does not want another tool, which costs time and energy.
- The current technical equipment in her practice could be better, but she can work with it and sees no added value in a new purchase.

GAIN POINTS:

- Personal contact with patients
- Problems and diagnoses can be discussed directly in person in an appropriate atmosphere
- A large group of patients who come to the practice on a regular basis

PAIN POINTS:

- Current management software solutions do not bring the relief hoped for
- Setting up and maintaining technical equipment is time-consuming and the company's own staff is not sufficiently trained for this purpose
- Great reservations when it comes to new technologies

CORE FEATURES:

- Support in the work process to save time is essential
- A system that combines all patient information and does not need to be maintained twice
- Secure rating system and regular review for unqualified ratings (or none)

Figure 10.20 Donna "The Diabetologist" Davis persona for Healthyfy.

Linda knows that she needs to get Donna excited about the platform. Otherwise, the product won't work because of the two-sided market problem. As a reminder, Healthyfy only brings value to doctors through patients and only brings value to doctors through patients, so both "parties" must be located simultaneously.

Therefore, Linda and her team are spinning various user stories around the topic of saving time at Donna. The main story is the following user story:

> *As Donna "The Diabetologist" Davis, I would like to have a way to be more time efficient with the exams and patient consultations in my practice so that I can help more patients in the same amount of time.*
>
> *The current inability to do this ensures we must turn away people who need my expertise daily.*
>
> *Enabling Donna to do this would lead to greater efficiency across the practice so that more people would know about Donna, benefit, and the practice would be even better economically positioned.*

After formulating user stories for Donna, the team decides to continue to focus there. The team is still uncertain whether the current idea brings sufficient value to physicians. Therefore, design solutions for the physician side of Healthyfy are to be created first. The approach: "If we can't develop a design solution that convinces physicians, we'll have to revise the idea altogether." That's why the team is turning to the most significant challenge first.

Chapter 11

Data-Driven UX Design in Detail: Developing Design Solutions

"Design is not just what it looks like and feels like. Design is how it works."
—Steve Jobs (entrepreneur and founder of Apple, among others)

The time has finally come to develop design solutions. This area is one that many people enjoy the most and is the core area that most people think of when someone says, "usability and user experience (UX) design." After reading those words, were user guidance, user interfaces (UI), and prototypes the first things that popped into your mind? That's great! Even though a large portion of what we present in this chapter is not design, everything should lead to design. User research, from our point of view, has the goal of developing great systems, products, and services. So, let's get started.

11.1 Reminder: What Is the Goal of Design, and How Do We Go About It?

The ultimate goal of design is, of course, to have a great product. Along the way, however, we want the ability to check multiple times to see if we might have taken a wrong turn. So, one design goal is also to enable evaluation with real users—in every iteration. To facilitate evaluation, you need to consider what the specific user interaction with your system, product, or service should look like in this phase. You should articulate the individual steps, the required information, and how you make information accessible to your users.

On the one hand, this accessibility is about good usability; on the other hand, you should also consider UX in the concepts. If you think back to the tree metaphor (see Chapter 2, Section 2.3), remember that ergonomics also exists in addition to usability and UX. Particularly ergonomics play a vital role if you design products with a physical component and not exclusively develop software. However, we'll only briefly mention the topic of *physical product design* in this chapter, as we assume that the majority of our readers are more interested in *digital product design*.

Following common principles for designing interactive systems is advisable when designing UIs. The interaction principles from International Organization for Standardization (ISO) 9241-11:2020 (presented in detail in Chapter 7, Section 7.4.4) are used in this chapter as an example.

One aspect to be considered in the design is the distribution of tasks between the user and the system. Technical systems are being automated to an ever-greater extent and can implement more and more tasks independently. However, when developing design solutions, also consider the following questions: “Where should the task be allocated? Should the users do the task, or should the system do it?”

Thereby the focus lies on the quality of execution, but also the “want” from the user’s perspective. As UXers, a best practice to advocate for those who consistently use the product. At this point, we often ask ourselves whether the task that remains with the user is still sensible—and, at best, even meaningful. If not, then the task should be automated as technically as possible.

Once you have determined who will take on which tasks, define the “how.” This question includes the specific design of the interaction (“Who, does what, when?”), but also defining how that happens: Is there a visual display, a sound, or a vibration? Is there a touch screen, a lever, or a brain-computer interface? In this step, you define all that and develop prototypes—often non-functional—that allow you to experience the interaction. We’ll now show you how to proceed step by step.

11.2 Design Research

We encourage you to start every design project with *design research*. Now, we can practically hear you screaming: “Research? Again? I thought we were finally getting started!”

Yes, we are. You know who the users are, their tasks, and the requirements for the design of the product and UI. That’s fantastic! Now, you just have to find out the current *state of the art* in terms of design.

Regarding *design research*, we don’t want you to spend hours on field research or interviews. However, we do recommend looking at four areas:

- Competitive solutions
- Solutions that address similar problems but in other sectors or industries
- The current *state-of-the-art design*, both in the business-to-business (B2B) environment and especially in the business-to-consumer (B2C) environment
- The existing range of owned products, solutions, and designs

Let’s look at these areas one after another.

Competitive Solutions

In this section of your design research, you get an idea of how the problem is currently being solved in design. You have probably already gained an initial insight into existing products and competitor solutions during your context of use analysis. Now, we look at these solutions in detail. In doing so, we recommend examining the existing solutions at various levels:

- Is it a product, a system, or a service?
- Does the solution consist of one application or several?
- Are there any accessories that are needed?
- How is the user guided through the solution?
- What is the design of the solution?
- For which values or associations does the solution or brand stand for?
- Are there patterns in the design that repeat between the different solutions?

This form of design research gives you a first impression of what users expect from your product. After all, you're usually not the first and only provider trying to establish yourself in this field. Thus, all previous experiences pay into the expectations of the users. So, knowing and understanding those previous experiences makes sense. Afterward, you can follow the design or do something completely different. Just be sure you make an informed decision.

Solutions That Address Similar Problems

Sometimes, you're inclined to look at a problem and say, "Oh, that's unique. No one has ever done that before." Believe us, that's practically never the case! In almost 100% of cases, comparable solutions will exist—either in your field, in which case we are looking at our competitors, or in another field.

For example, suppose you want to develop a patient monitor for the medical sector that measures all live data on the patient's condition and displays it clearly. In this case, you'll probably first look for comparable patient monitors—obviously. But afterward, you might broaden your view: Are there other applications in the healthcare industry where live data is also measured and displayed together? You might then come up with systems that monitor entire wards. Or you might think of other measurement systems, such as ultrasound systems. For each of these systems, you can ask yourself the following questions:

- What from the other design is applicable to my problem?
- What from the other design, unfortunately, does not fit?
- What can I learn from their implementations for my own solution?

Afterward, you can abstract even further by asking the following questions:

- Where do live measurements and the display of data take place outside of medicine?
- How is the data displayed?
- What are the similarities, and what are the differences?

Then, for example, you could develop the idea that process controls in manufacturing plants should also collect and display live data.

This step aims to get a feeling for various aspects of design, such as the following:

- The color language of the relevant industry
- The language of form in a branch
- Certain display sequences in a branch (chronological sequence of information)
- Certain display structures (setups) in an industry
- Extended information, which is additionally provided
- Successful design solutions (from other areas) for your goal
- Unsuccessful design solutions (from other areas) for your goal

In this thought exercise, you'll get a good feeling for what is out there, what makes the industry tick, and what else might work great. You'll also get a feeling for the opposite: Which solution approaches do not suit you, and what do you definitely not want for your system, product, or service?

Current State-of-the-Art Design

Another area you should look into is the current state of the art, i.e., the current state of development—once in your industry, but especially in the B2C sector, regardless of your solution.

Good starting points for this can be widely used solutions such as smartphones or televisions. Why? Because almost every user here knows the current state of development and uses this as a reference. In the US, around 307 million people used a smartphone in 2022, around 92% of the entire population (*https://statista.com*). So, you can firmly assume that most people you design products for are familiar with current operating systems and designs on their smartphones and use them as a comparison. This is even more true if your system, product, or service is a digital solution.

Only if you explicitly design for a significantly older target group should you expect users with less experience with smartphones. Considering the total number of 282.1 million smartphone users in relation to the approximately 332 million inhabitants in the US, please do not forget that over 27 million children under 12 live in the US alone. Don't assume that the missing 23 million users are exclusively older people.

The Existing Range of In-House Products, Solutions, and Designs

The final area you should consider during the design research is the existing in-house product and design palette. The focus lies on understanding the design language of your company and your product family and helping develop or refine your design. In addition, if a product family appeals to the same user group, the products should also feel similar.

From a business perspective, your solutions should also speak a common language. Good sources for this shared vocabulary are often handbooks, the products themselves, and websites or existing style guides. Use everything possible to get a sense of the look and feel of your company and products.

Documentation of the Design Research

You should compile and document all these findings. The goal of design research was initially to get an overview and create a common ground for further decisions. Typical questions include how to place your solution and further develop the brand. Other stakeholders are also involved in these decisions. Clean documentation of the design research helps to make these decisions.

Design research documentation, however, is not about a particularly complicated structure. In the simplest case, you have a presentation divided into the four areas described earlier, and you simply insert images, screenshots, and examples. The only important thing is having the information ready for when you get down to the visual design. However, we're still far off from this step (Section 11.12). At this point, let's start with the basics.

11.3 From Coarse to Fine

In most design projects, the solution is no longer completely free but already predefined due to preliminary work or stakeholder desires. So, you may start the design process knowing that the solution will be a smartphone app. Or you're a service provider for smartphone app development, and even then, what type of solution will come out is usually predetermined. However, you sometimes start "off the beaten path" with a new product.

If you're starting from scratch, a good starting point for design is *brainstorming* or *brainwriting* (see Chapter 4, Section 4.2.4, for a description of both methods), where you gather initial ideas for the possible solution. At this point, we're still at an elementary level and can think of different approaches to the solution. Then—in the course of the future iterations—you'll further commit to a design and a solution and flesh it out in more and more detail and depth.

We recommend always working from coarse to fine when designing your solution and finding the right design. First, define the overall structure before working on the concrete content, icons, colors, and fonts.

Why? In our experience, you can easily invest a lot of time in visual design and forget that you must also think about workflow optimization and comprehensibility. In this section, we'll provide you with approaches that work well for thinking about a solution from the ground up from the user's perspective.

When we start development, we usually have two strands going on in parallel: on the one hand, the *development of the workflow* and the individual user interactions. That strand is the coarse part where we are mainly concerned with optimizing processes (i.e., reducing operating times, errors, and clicks). On the other hand, the *development of the visual design* usually runs in parallel but later. This strand is the fine part.

As mentioned throughout this book, we decided to separate the two strands in content because assessors often find it challenging to separate design from the structure in user testing. In a *user test*, if you show a visually designed UI, you'll often get feedback mainly on the visual appearance: If you don't like its visual appearance, you won't like the whole product. If you like the look, then you could like the entire solution. Thus lies the danger that you won't even get down to the level of the operating processes—because you can't dive that deep. So, if you move to the fine level, you can't optimize the product's usability, which is why we would like to devote our undivided attention to this area.

11.4 Developing Basic Solution Options

Often, you start the entire data-driven UX design (3DUX) process with an idea of how to solve the problem. In this case, you run the risk that the existing solution in your head may not be the best. Nevertheless, we recommend entering the basic solution development process with a solution in mind. If the solution you have already thought of crystallizes, all the better. Above all, you can clearly explain to stakeholders why you are (or are not) pursuing a particular solution.

In practice, often a problem is identified for which a (clear) solution cannot yet be derived. In this case, too, our recommendation is to develop basic solution options.

In both cases described, we assume you have already understood the problem and the basic requirements through the first two steps in the data-driven UX design process (context of use analysis and deriving usage requirements). When you now start looking for a solution, we recommend beginning in the spirit of *design thinking* (see Chapter 4, Section 4.2).

Gather a team of people with different expertise and backgrounds who are connected to the problem or will be involved in solving it. Clarify the current status beforehand or at the beginning so that everyone in the team knows the framework. Everyone should thus know in advance what information has been gathered so far, including the design research.

Our experience shows that the brainstorming or brainwriting that follows works best when the participants are prepared, raising the overall quality of the ideas. Perhaps the participants even come to the joint workshop with their own initial ideas. In this case, the important focus is that the workshop's goal is to develop new ideas and evaluate existing ideas, not to defend one's own ideas.

And then, you can brainstorm ideas together in a brainstorming session or separately in a brainwriting session. The ideas will still be raw and unpolished at this point in the design process. But you'll already be able to crystallize initial basic tendencies. Is it an app, a web application, or a platform? Is it a physical product? Do we want to implement a service? Then, grab all the individual ideas, discuss them, refine them a bit more, and go straight into the evaluation.

Evaluating ideas can take an infinite amount of time. As described in Chapter 4, Section 4.2, you could design a matrix to help you evaluate every requirement for every idea. If you try this, however, you'll quickly realize that you do not know enough about the individual ideas to decide which requirements are fulfilled and which are not.

The second procedure, also described in Chapter 4, Section 4.2, helps with this, which you can use to conduct a "quick and dirty" evaluation of all these ideas: the three intersecting circles.

As a reminder, divide your ideas into the categories shown in Figure 11.1. (These categories are merely suggestive.)

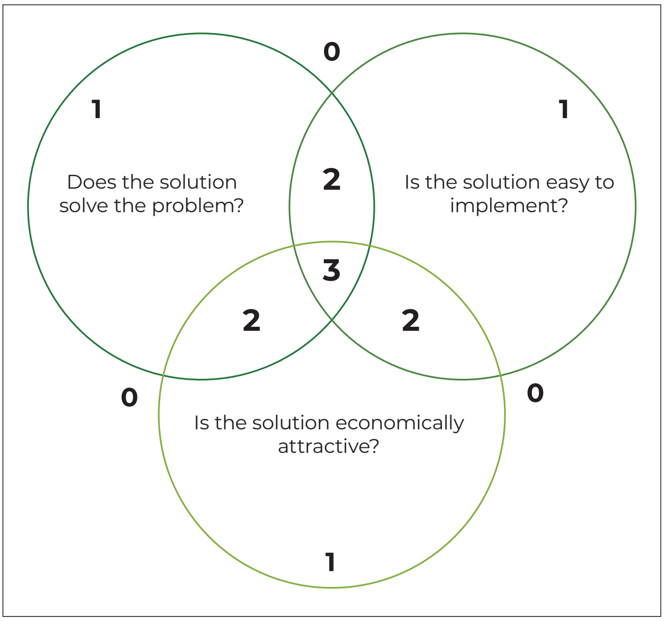

0
1
1
Does the solution solve the problem?
2
Is the solution easy to implement?
3
2
2
0
0
Is the solution economically attractive?
1

Figure 11.1 Dividing your ideas into the three circles or categories for an initial evaluation.

In our brainstorming and brainwriting sessions so far, we rarely only had one idea in the 3-point region (i.e., one idea met all criteria). In significantly more cases, we either had no ideas with 3 points or extremely many, if the team was very well prepared (hence our recommendation on preparation from the beginning of this section). Again, we would like to give you more tips on proceeding in these situations.

11.4.1 How to Turn a 2 into a 3

If you don't have any 3-point ideas, the best option is to look at all 2-point ideas individually. According to our definition, a 2-point idea fulfills two out of three relevant criteria. So, if you now take a 2-point idea, you know, based on the position in the circle, that the idea solves the problem and is easy to implement, but unfortunately, not economically interesting. At this point, start a mini-brainstorming session and ask yourself, "How could we make this idea commercially interesting?" If you have an idea for that: Wonderful! Add it to the middle of the three circles—you've just created a 3-point idea. If you and the team can't think of anything: No problem. Just take the next 2-point idea and check if you can implement the missing criterion.

11.4.2 How to Deal with Too Many 3-Point Ideas

If you have too many 3-point ideas, you're spoiled for choice. Further selection may be necessary to focus the team and move forward more quickly. For us, two different approaches have proven promising in this situation:

1. You do a score poll.
2. You do another assessment, this time with scales.

The *score poll* is much faster, more pragmatic, and offers much less room for discussion. For this poll, simply hand each team member three sticky dots (or digitally allocate three votes) and have them place dots on the idea or ideas they find most promising. We usually also allow all three dots from one participant to be stuck on the same idea. Afterward, count the points and choose the one or two ideas with the most votes.

If you want to cover a special *weighting by specific team members*, you can facilitate this approach either by giving more points to the respective person or by using different colored points, which then have a different weighting. It would be best if you also removed any ideas in the beginning that were vetoed by a decision-maker who is present.

We know this exercise is a controversial proposal because excellent ideas could fall between the cracks. Maybe even the best idea could die at the beginning. But our experience shows that ideas, with a *decision-maker veto*, are usually frozen and stopped at a later stage anyway—only by then, significantly more resources have already been used. Use your sensitivity and knowledge of the decision-makers to consider whether the risk exists.

Scoring is a quick and easy way of weighting. But sometimes, the decision-making feels too fast and imprecise. In this case, *scoring using scales* is a good option. For all 3-point ideas, you have already decided that all three criteria are met. But taking a closer look, you'll find that each idea fulfills the criteria differently. To get more into numbers instead of relying on gut feelings, you can also evaluate the individual aspects on a scale of 10 so that, in the end, you can locate each idea in a (three-dimensional or several two-dimensional) coordinate system(s).

The introduction of the 10-point scale makes it easier to distinguish between ideas and to get a feeling for whether a criterion is met to a slight, considerable, or high degree. But beware: this feeling is only a pseudo-certainty. After all, no implementation exists yet, and you still know very little about each idea. Thus, don't dwell too long on the evaluation, but try to arrive at a rating quickly.

11.4.3 Conclusion

Utilizing brainwriting, combined with the three circles and scoring methods, you can identify an initial favorite idea for elaboration within 2 to 4 hours (provided you have an accomplished facilitator to guide the team back to the topic).

Note at this point that there are, of course, many other methods that can help you evaluate ideas. In this book, we've only presented the method we found to be the fastest and most effective for usability and UX design work. *Speed* is also the main reason for our choice: Since you're evaluating theoretical ideas at this stage and still know little, we recommend learning quickly rather than slowly.

Thus, at the end of this section, you should have selected one or more promising basic ideas from a wide range of ideas, and you can start fleshing them out. An excellent way to flesh out these ideas is with prototypes.

11.5 Working with Prototypes

By *prototype* in the field of data-driven UX design, we mean a fast and simulated implementation of the solution idea. We agree with all approaches that do not necessarily require an executable version but rather have the goal of mapping the interaction between the user and the product as quickly and simply as possible.

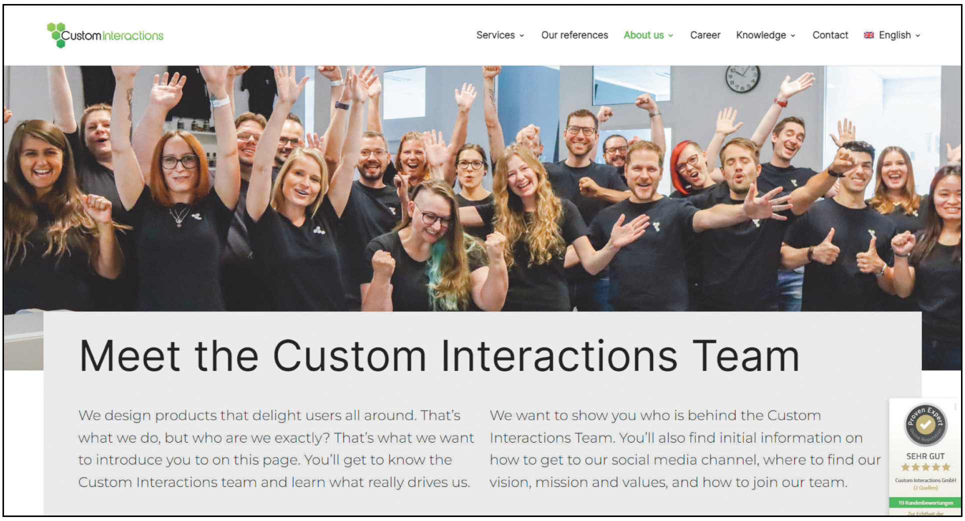
Depending on your approach, your prototype can be anything from a block model to a cardboard model, a drawing, a video, or a digital prototype you made yourself. Once you decide to work with a prototype, you must determine beforehand what you're building a prototype for. Depending on the goal, you'll need to fulfill different requirements for the prototype, and the effort required to create it will change.

Therefore, an essential step is to think about the goal in advance. In this section, we'll explore the different types of prototypes—and then return to our discussion of the goal.

11.5.1 Horizontal and Vertical Prototypes

In general, you can distinguish between two kinds of prototypes: horizontal and vertical. *Horizontal prototypes* present the solution in its full breadth; that is, they show the possible range of functions without defining the individual interaction paths. They go into the breadth—the horizontal—but not the depth. An example of this could be a home page on a website where the complete navigation menu is there with all the entries, but none are clickable, as shown in Figure 11.2.

The opposite of this are *vertical prototypes*, which usually do not provide an overview of the solution, but make a specific workflow testable instead. Following a particular workflow, these prototypes go deep (i.e., vertical). The breadth of the solution is neglected.



Meet the Custom Interactions Team

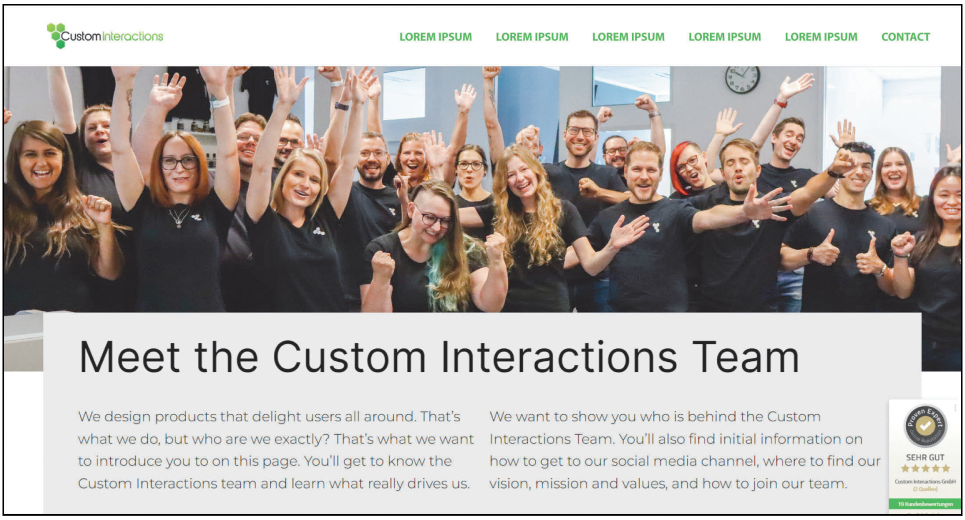
We design products that delight users all around. That's what we do, but who are we exactly? That's what we want to introduce you to on this page. You'll get to know the Custom Interactions team and learn what really drives us.

We want to show you who is behind the Custom Interactions Team. You'll also find initial information on how to get to our social media channel, where to find our vision, mission and values, and how to join our team.

Figure 11.2 Horizontal prototype of the website "www.custom-interactions.com." The page is almost entirely worked out (including the visual design). However, none of the links are clickable.

An example could be the same start page, which until now has only placeholder texts (**LOREM IPSUM**), except for one link. However, when you click on that link, you experience the entire interaction to completion instead, as shown in Figure 11.3.

Of course, you can also combine both variants. However, if you build a completely defined prototype that describes every step in terms of both breadth and depth, you might be spending too much time on the prototype.



Meet the Custom Interactions Team

We design products that delight users all around. That's what we do, but who are we exactly? That's what we want to introduce you to on this page. You'll get to know the Custom Interactions team and learn what really drives us.

We want to show you who is behind the Custom Interactions Team. You'll also find initial information on how to get to our social media channel, where to find our vision, mission and values, and how to join our team.

Figure 11.3 Vertical prototype of the website "www.custom-interactions.com." The menu on the start page contains various placeholders. However, a single workflow (contact) is fully designed.

11.5.2 High- and Low-Fidelity Prototypes

Whether you have invested too much time in a prototype also depends on how much effort it took to create the prototype. In this context, too, different types of prototypes can be distinguished: low-fidelity prototypes and high-fidelity prototypes. They differ in the level of detail in the design and the options already built in. Note that "fidelity" is not a yes/no question, but exists on a scale. Some prototypes will fall in the middle range. But what does that mean exactly?

A *low-fidelity prototype* is exclusively about making the interaction and mode of action clear. Hand drawings, modeling clay, simple pieces of paper, or other gadgets can be used for this purpose, as shown in Figure 11.4.

Low-fidelity prototypes can usually be produced extremely quickly and cost little money, which makes them ideal for getting a first impression of the quality of the solution. In addition, their rustic appearance ensures that the evaluation focuses on the solution and the interaction and does not drift too quickly into the visuals: Every evaluating person looking at a low-fidelity prototype can clearly see that this is not the final product. This clarity is a huge advantage when you want to talk about the interaction and the mode of action—and not yet about the visuals.

Moreover, all too often, hours, days, and weeks are invested in tweaking the look of the wrong product. We discussed this issue earlier in Section 11.3 and would like to reiterate it here. We always start with low-fidelity prototypes such as *wireframes* (Section 11.6) and *screen flows* (Section 11.8) before moving on to high-fidelity prototypes.

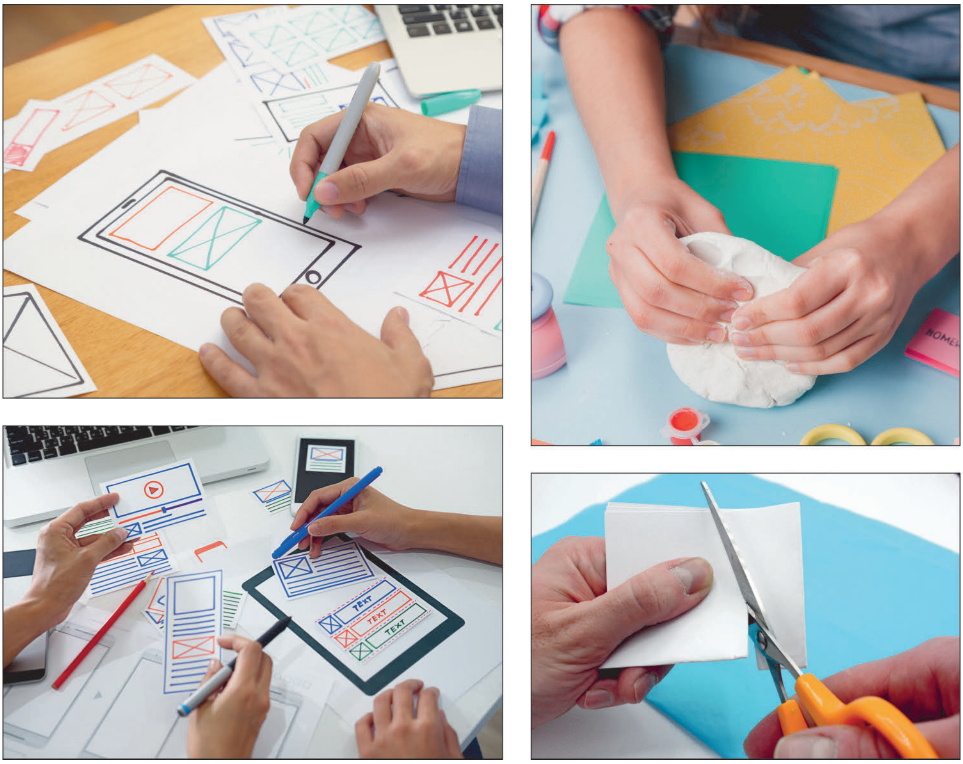


Figure 11.4 Collage with different low-fidelity prototypes. What they all have in common is that they illustrate the interaction and the mode of action.

High-fidelity prototypes are based on the appearance of the finished product. It may even be the case that a skillfully made prototype is indistinguishable from the finished product, as shown in Figure 11.5.

A high-fidelity prototype is where the mode of action, the interaction, and the design meet. The advantages are obvious: With high-fidelity prototypes, you can evaluate all aspects of a design without actually having implemented the solution. On the other hand, high-fidelity prototypes take much longer to produce and are usually more cost-intensive, so they are only useful in some instances.

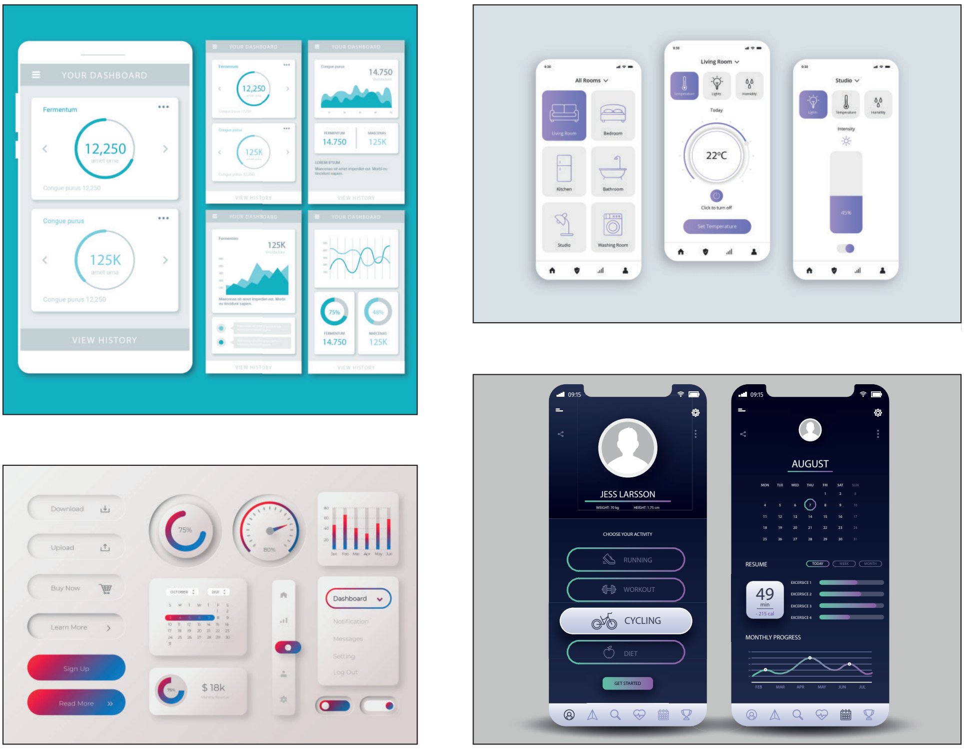

YOUR DASHBOARD
Fermentum
12,250
Congue purus
125K
VIEW HISTORY
14,750
All Rooms
Living Room
Bedroom
Kitchen
Bathroom
Studio
Washing Room
22°C
Set Temperature
Download
Upload
Buy Now
Learn More
Sign Up
Read More
75%
80%
$ 18k
Dashboard
Notification
Messages
Setting
Log Out
JESS LARSSON
CHOOSE YOUR ACTIVITY
RUNNING
WORKOUT
CYCLING
DIET
GET STARTED
AUGUST
RESUME
49
MONTHLY PROGRESS

Figure 11.5 Collage with various high-fidelity prototypes. What they all have in common that they already look like a finished product.

11.5.3 Which Prototype Should Be Used and When?

Before we create a prototype, a goal must be defined. Based on this goal, the choice of the prototype usually results relatively directly.

For example, if a specific workflow is to be tested with users in user research, we need a vertical prototype that maps the workflow. If we want to fundamentally test whether the solution is going in the right direction, then a horizontal prototype is a good choice. This combination is also often used in user tests. You start in a horizontal area (e.g., with a start page or a dashboard) for the overview and then go vertically deeper into a workflow with the users.

The choice according to fidelity is also based on the goal. We use high-fidelity prototypes, for example, when the prototype is used for advertising purposes—in other words, when internal or external stakeholders need to be convinced of a product. In this context, technical aspects often play a much smaller role than the UX of the decision-makers, which is why the effort is worthwhile (also, the effort is still far less than developing the final product). Another example could be a trade show, where you want to show something about the future product to collect customer feedback.

For our projects, however, low-fidelity prototypes in which particular areas are visually designed are often sufficient. We use the low-fidelity prototypes for feedback on the idea and the interaction; we then use the designed area to gather feedback on the visual appearance. In this way, we achieve a perfect balance between feedback on the interaction concept of the application and the visual appearance and UX. Furthermore, this approach is faster and more cost-efficient than a "full" high-fidelity prototype.

For a better understanding, you can map the different prototype types in a 4-field matrix, as shown in Figure 11.6.

On one matrix axis, you have the fidelity of the prototypes, and on the other axis, you have their alignment. You don't need to fill in this matrix completely, but you can use the matrix to think in advance about what kind of prototype you need. This consideration will help you invest only the effort required for your endeavor. This book assumes that you're probably working on a digital offering. Therefore, in the next section, we'll introduce you to a special low-fidelity prototype that plays a significant role in the digital world: the wireframe.

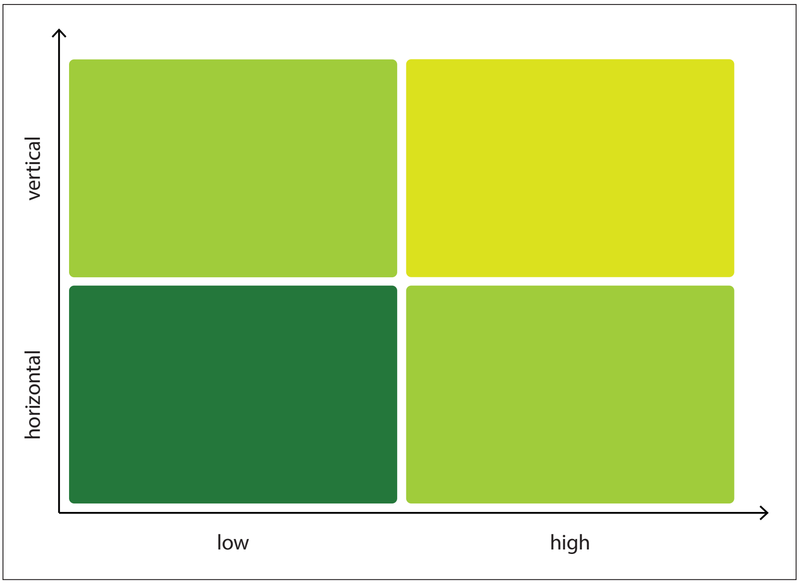


Figure 11.6 Four-field matrix for systematizing prototypes.

11.6 Wireframes

The word *wireframe* can be taken in a pretty literal sense. It means that the basic structure of the screen is represented schematically. Usually, a wireframe does not yet contain any visual design but is limited to the representation of the individual elements in relation to each other, as shown in Figure 11.7.

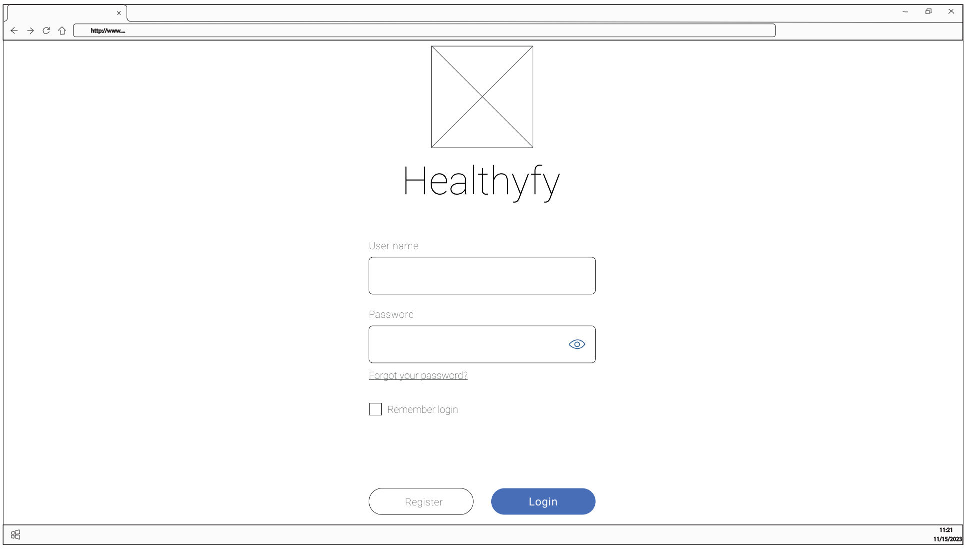


Figure 11.7 Wireframe of the logon screen for Healthyfy. Notice how most elements consist of placeholders or outlines. With a few exceptions, there is no use of color yet.

Wireframes make it possible to easily map a basic structure of the digital application and later evaluate it by users. As shown in Figure 11.7, notice that the most crucial interaction elements can be seen on this screen. In this example—but also in wireframes in general—graphics are only indicated by placeholders. For instance, in the place where the Healthyfy logo would later appear, there is only a square with diagonal lines: a typical placeholder in wireframes.

When creating wireframes, several things need to be clarified in advance:

- For which device are you designing the wireframes?
- What resolution should you use for the wireframe?
- What specifications for the wireframe already exist?
- What content do you want to present first?

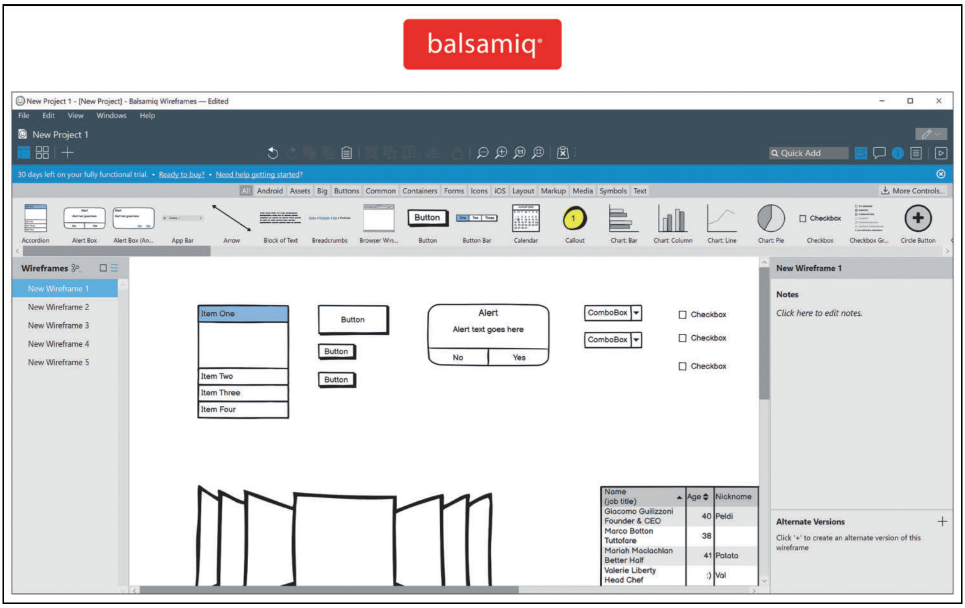
Before creating a wireframe, you should be clear about the device for which you're making a wireframe. Since the display or operating device considerably influences the design of the wireframes, the resolution, existing defaults, and content, you should clarify this question first. If you haven't decided yet, you can also try out different types of devices. To make your choice easier, Section 11.10 will help you decide on the device. In this section, we'll provide some advice on typical resolutions.

Once you have decided on a device and a resolution (even only for the first iteration), now is the time to create your workspace in the appropriate size. Doing so will nip design problems in the bud that might need additional adjustments later in the implementation if the aspect ratios weren't thought out correctly. To create the wireframes, we usually work with either *Adobe XD*, *Axure RP*, or *Figma*, as shown in Figure 11.8 and Section 11.9.1.

Adobe XD
Axure RP
Figma

Figure 11.8 Prototyping tools we use: Adobe XD, Axure RP, and Figma.

But of course, you can also use any other prototyping tool. *Microsoft PowerPoint* or *Balsamiq* are also perfectly suitable for creating wireframes. The advantage of the first set of tools is that you can also create high-fidelity prototypes in the same tool. Thus, if you also want to create a visual design, we advise using Adobe XD, Axure RP, or Figma. However, we recommend Balsamiq if you have little experience or come from a visual design background, as shown in Figure 11.9.


balsamiq
New Project 1 - [New Project] - Balsamiq Wireframes — Edited
File Edit View Windows Help
New Project 1
Quick Add
30 days left on your fully functional trial. • Ready to buy? • Need help getting started?
All Android Assets Big Buttons Common Containers Forms Icons iOS Layout Markup Media Symbols Text
More Controls...
Accordion Alert Box Alert Box (An... App Bar Arrow Block of Text Breadcrumbs Browser Win... Button Button Bar Calendar Callout Chart: Bar Chart: Column Chart: Line Chart: Pie Checkbox Checkbox Gr... Circle Button
Wireframes
New Wireframe 1
New Wireframe 2
New Wireframe 3
New Wireframe 4
New Wireframe 5
Item One
Item Two
Item Three
Item Four
Button
Button
Button
Alert
Alert text goes here
No
Yes
ComboBox
ComboBox
Checkbox
Checkbox
Checkbox
New Wireframe 1
Notes
Click here to edit notes.
Alternate Versions
Click '+' to create an alternate version of this wireframe

Name (job title)	Age	Nickname
Giacomo Guilizzoni Founder & CEO	40	Peldi
Marco Botton Tuttofare	38	
Mariah Maclachlan Better Half	41	Patata
Valerie Liberty Head Chef	:)	Val

Figure 11.9 Balsamiq is a pure wireframe tool, which has the advantage that you cannot "get lost" in the visual design, but also a disadvantage in that you cannot visually design once you have left the wireframe state.

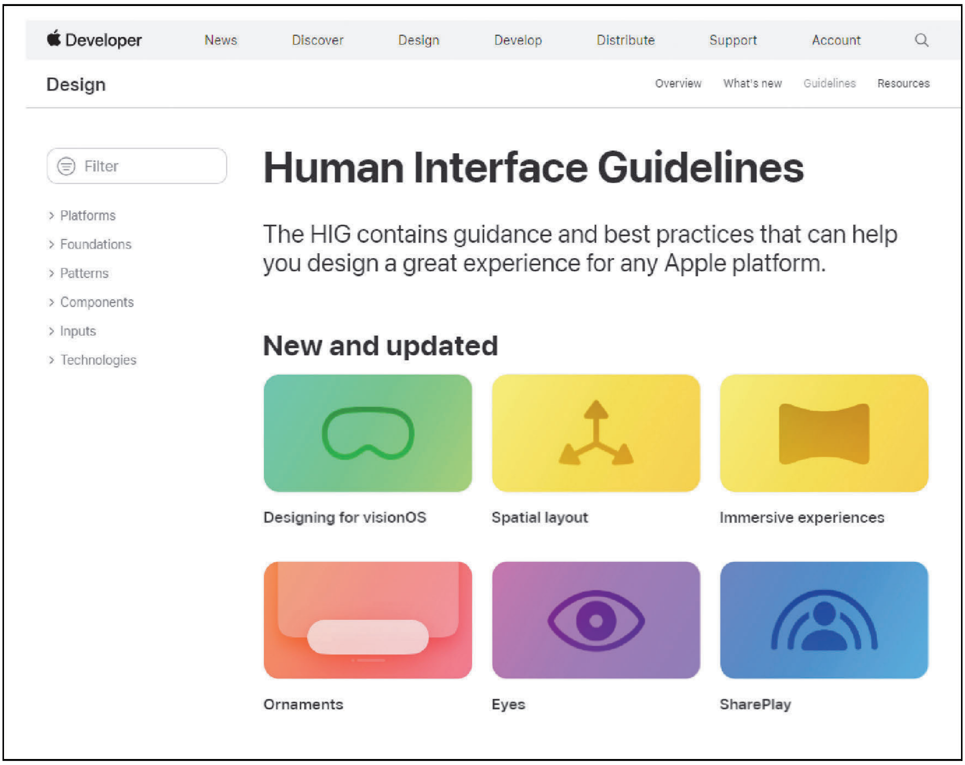
Balsamiq then has the advantage that you cannot design visually. Thus, you are forced to create a wireframe and can't lose yourself in the visuals as much as with the other tools.

Figma and Adobe XD offer the additional advantage that work can be done cooperatively on a project, and everyone can see the same status in real time. Especially for distributed teams, this feature can be a great added value in the design.

Suppose you have decided to design *paper wireframes* or *drawings*. You should ensure that the aspect ratio fits your drawing area before you start to avoid unpleasant surprises later.

In addition to selecting the correct device and an appropriate resolution, usually other specifications will require attention. These specifications can come from existing style guides or element libraries from your company or the client and, for example, might define menus or interaction areas. But they can also be general interface guidelines for an underlying operating system, for example, if you design for iOS or Android.

Apple, for example, offers its developers *Human Interface Guidelines* that define how information must be presented and which elements can be used so that one's application fits ideally into the existing design world, as shown in Figure 11.10.


Developer
News
Discover
Design
Develop
Distribute
Support
Account
Design
Overview
What's new
Guidelines
Resources
Filter
Platforms
Foundations
Patterns
Components
Inputs
Technologies

Human Interface Guidelines

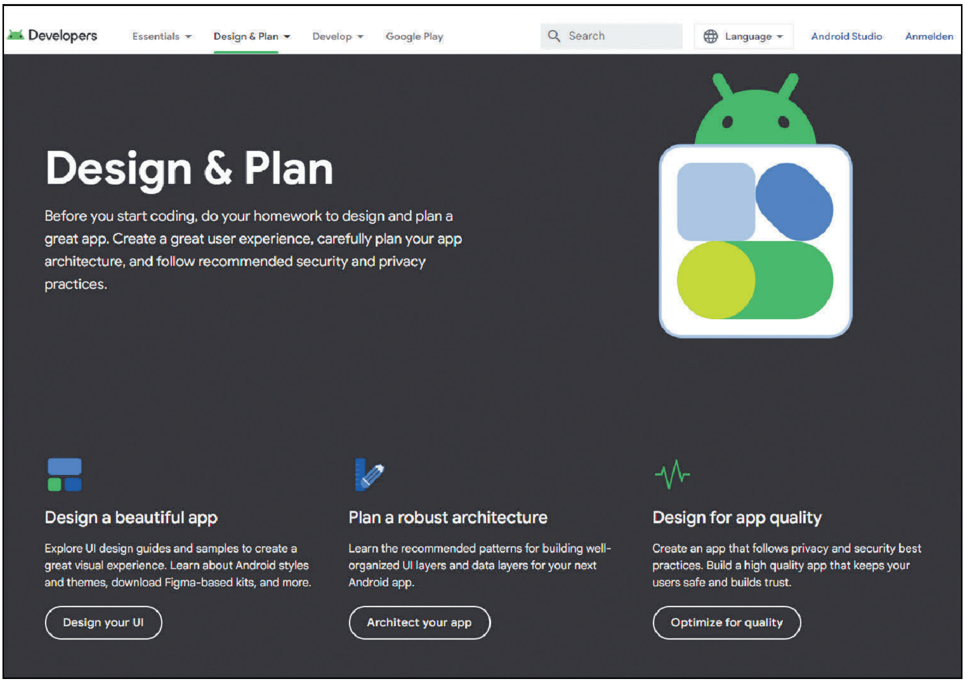
The HIG contains guidance and best practices that can help you design a great experience for any Apple platform.

New and updated

Designing for visionOS
Spatial layout
Immersive experiences
Ornaments
Eyes
SharePlay

Figure 11.10 Apple's Human Interface Guidelines at https://developer.apple.com/design/human-interface-guidelines.

The same or similar information can also be found for Android. In addition to the *Material Design Guidelines*—as the style guide for Android is called—a collection of material design elements is available that can be used directly to build an interface that conforms with the style guide, as shown in Figure 11.11.



Design & Plan

Before you start coding, do your homework to design and plan a great app. Create a great user experience, carefully plan your app architecture, and follow recommended security and privacy practices.

Design a beautiful app

Explore UI design guides and samples to create a great visual experience. Learn about Android styles and themes, download Figma-based kits, and more.

Plan a robust architecture

Learn the recommended patterns for building well-organized UI layers and data layers for your next Android app.

Design for app quality

Create an app that follows privacy and security best practices. Build a high quality app that keeps your users safe and builds trust.

Figure 11.11 The design guidelines for Android at https://developer.android.com/design.

If you design for a well-known brand or system, you should assume that these specifications exist in one form or another, either directly or through third-party providers. We rarely see a case where the design is completely free and undefined. The only exceptions to this rule are when building a new brand or rebuilding an existing brand. In all other cases, specifications must usually be considered. Knowing these specifications will make the design easier because you can build upon them. Therefore, at the very beginning, try to grasp the framework conditions as clearly as possible to save time and effort.

Once you have clarified all the framework conditions, you still have one last decision to make: What do you want to present first regarding content?

This question is where you get into the *content consideration*. We usually start by defining the most important workflow for the user. By *workflow*, we mean which steps must be taken so that a particular action happens. A rough definition of this workflow can be made at this point.

For example, imagine we're talking about our Healthyfy example. You might define the main workflow from the patient's perspective with the following steps:

1. Search for the app in the app store
2. Download the app
3. Open the app
4. Register with Healthyfy
5. Search for a specialist
6. Make an appointment with the specialist
7. Enter an appointment reminder in the calendar

You can represent the workflow in writing or as simple boxes, as shown in Figure 11.12. At this point, the only essential task is to clarify what the *user journey*—the user's path through the application—should look like. And by "look," we don't mean visually, but in terms of content. What steps are necessary for the user to perform this task?

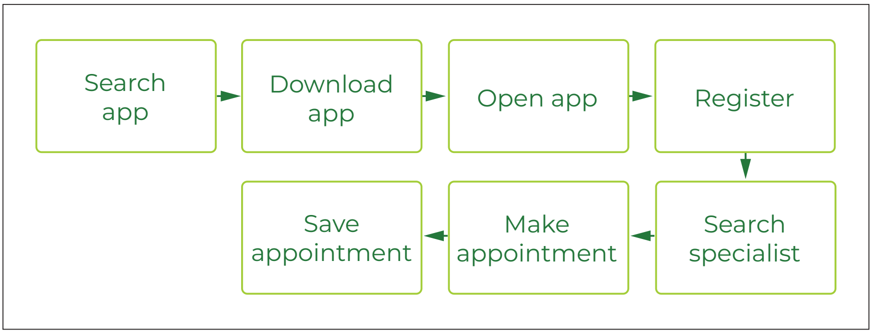


Figure 11.12 The main workflow for Healthyfy, shown as simple boxes.

As described in Chapter 5, Section 5.2.3, we recommend that you look at the risk of each workflow and how often it is used when selecting which workflows to design first. Start with the workflows that are high risk and frequently used. This focus will ensure you can get feedback on the most important workflows as early as possible.

Now, you can start wireframing. Think about how the wireframe should look for each step or for each box. In the previous example, you don't need your wireframe for the app store search or download because external apps predefine these pages. You can therefore use these elements for the time being.

Next, you should start thinking about what the first page of your app might look like when a user opens that app. At this point, you can start building individual wireframes. In contrast to the wireframe showing Healthyfy's logon screen on a desktop web browser (shown earlier in Figure 11.7), the app's first screen might look like the wireframe shown in Figure 11.13.

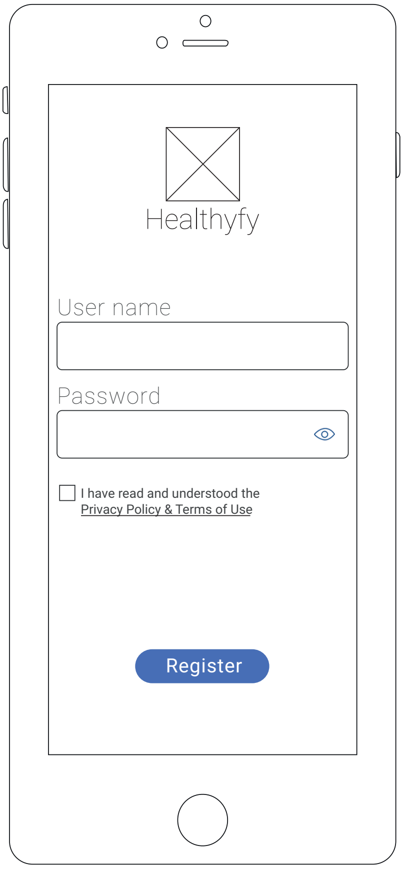

Healthyfy
User name
Password
☐ I have read and understood the
Privacy Policy & Terms of Use
Register

Figure 11.13 First app screen for the main workflow for Healthyfy.

As the vocabulary may indicate, creating wireframes is a structured process. This activity is often performed by *concept designers* (people who specialize in creating UI concepts). Wireframes can be largely composed of existing elements that map the interaction piece by piece. At first glance, using ready-made elements to create a UI may not seem very creative—and rightly so, because it limits your design.

On the other hand, using premade elements gives you an enormous advantage when converting your wireframes into a real product since not all elements will need to be created individually but instead can be used from a *standard library*.

From time to time, of course, you may still find it necessary to create your own interface elements or to combine several interface elements to achieve a better UX. However, before you resort to in-house development, you should first check if a suitable component is already available. Keep in mind that this availability does not predict how the later visual design of your app will be performed. Even using standard components, you can create individual and creative solutions.

But how do you know which components are suitable for what? Clear recommendations for these choices now exist, and even an ISO standard deals with questions regarding which element should be used and when. This standard is ISO 9241-161:2016 and is called “Ergonomics of human-system interaction—Part 161: Guidance on visual user interface elements.”

In the following sections, we’ll present individual elements to help you select the correct elements for designing your wireframes. We won’t present all the possible components but have selected the most common ones. We recommend you look at the standard if you’re interested in completeness. In most cases, we’ll also resort to a wireframe representation when presenting the elements. However, if the explanation should be more accessible with another representation (e.g., with color), we won’t stop ourselves and will show you other illustrations besides wireframes.

11.6.1 Buttons

One of the best-known interface elements is the *button*. You use a button whenever you want to trigger an action—for example, leaving a page, saving information, or submitting a form. Buttons appear in different variants and shapes, which we summarize under the term “button.” Please forgive us as we reduce the complexity of the topic a bit.

In its simplest form, a button in a wireframe is simply a labeled rectangle, as shown in Figure 11.14.

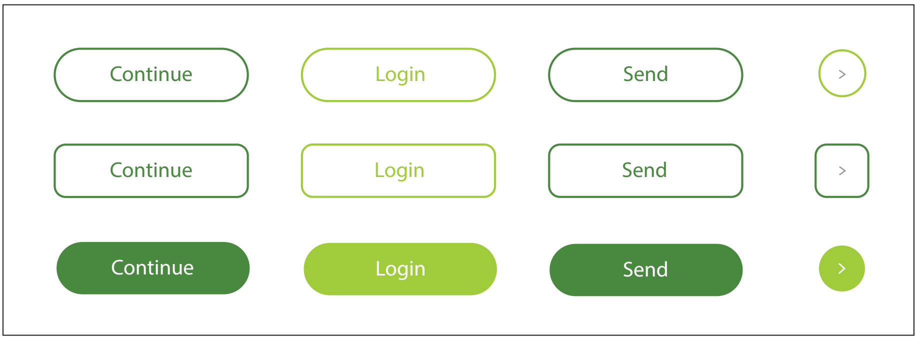


Figure 11.14 Various wireframe buttons.

So that you can easily recognize this shape in the wireframe as a button that can be pressed, buttons are usually combined with a label and/or an icon. The label or icon usually shows the action triggered by clicking the button. A typical button could be labeled **Continue**, **Login**, or **Send**, as shown in Figure 11.15.

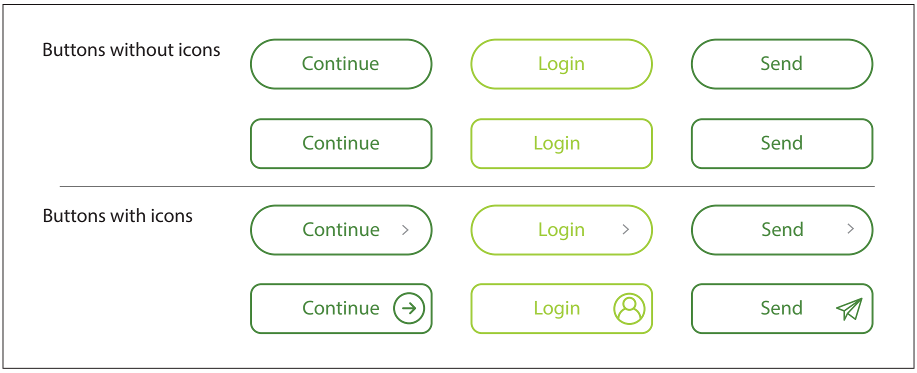

Buttons without icons
Continue
Login
Send
Continue
Login
Send
Buttons with icons
Continue
Login
Send
Continue
Login
Send

Figure 11.15 Labeled wireframe buttons with and without icons.

Typical examples of buttons with icons are **Download** or **Save**, as shown in Figure 11.16. In this case, the icons used have become so widespread that, in most cases, an explanation is no longer necessary—even if the floppy-disk origins of the **Save** icon are unfamiliar to newer generations.

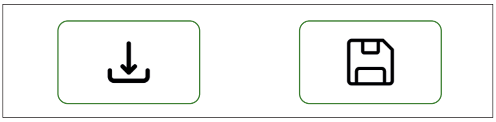


Figure 11.16 Wireframe buttons with icons and without font.

Practically all digital interfaces contain buttons in some form. In many cases, buttons stand alone and are used to navigate forward and backward in a workflow. In other cases, many buttons appear together as part of a toolbar, as shown in Figure 11.17.

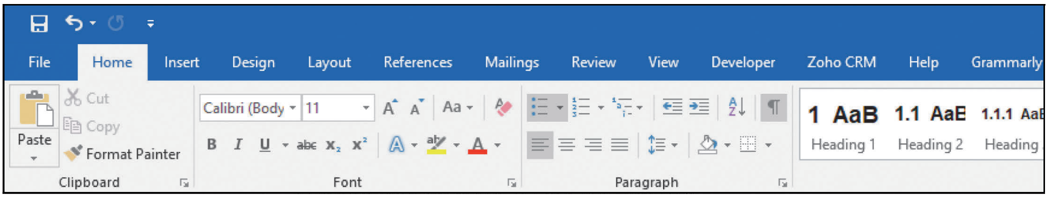

File
Home
Insert
Design
Layout
References
Mailings
Review
View
Developer
Zoho CRM
Help
Grammarly
Cut
Copy
Paste
Format Painter
Clipboard
Calibri (Body
11
Aa
Font
Paragraph
1 AaB
Heading 1
1.1 AaE
Heading 2
1.1.1 AaE
Heading

Figure 11.17 Buttons in various forms from the word processing software Microsoft Word (not a wireframe but a screenshot).

As part of a toolbar, buttons are often used to call up further dialogs, which contain buttons that can be used to continue or cancel the dialog. In case buttons are used to navigate through a process, buttons that lead to the next step are often positioned at the bottom-right or top-right corners in Western cultures. Buttons that cancel the process or go back one step are often placed in the lower-left or upper-left corners. This approach creates a natural mapping to progress or regress, making it easier for users to move through the dialog quickly, as shown in Figure 11.18.

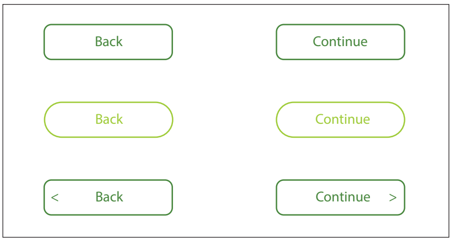


Figure 11.18 Examples of the arrangement of “Continue” and “Back” buttons.

In cases where the action may result in a critical outcome, such as deleting input or content, it may be useful to swap the placement of the buttons so that the **Cancel** button is on the right. The button for confirming is on the left, as shown in Figure 11.19. This way, in case of a “blind click,” a safe mistake is made, and no data is lost. Alternatively, a downstream confirmation can be used.

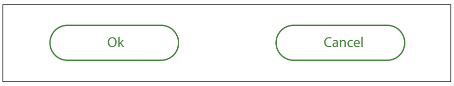


Figure 11.19 Arrangement of the “Cancel” button on the right. Thus, the safe action is triggered when “blindly clicking.”

11.6.2 Dropdowns

Dropdowns are fields that allow selecting an option from a set of given alternatives. A typical use case for a dropdown would be selecting a hoodie size while shopping online, as shown in Figure 11.20. When sending a package, a list of countries as destination selection is also a typical example of a dropdown—although a more suitable element is available in many situations (i.e., the *combo box*). We’ll look at the combo box in Section 11.6.5 in more detail.

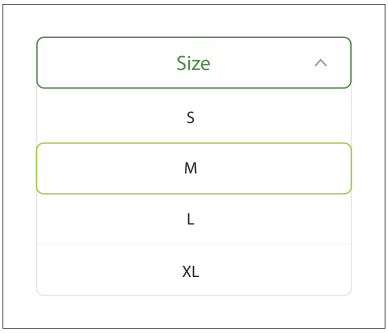


Figure 11.20 Typical dropdown for selecting an option from a fixed number of alternatives.

Dropdowns have an advantage in that the field requires little space since only one answer is visible before and after the selection. So, with the use of dropdowns, you can achieve an interface that remains clear and concise, although (many) different alternatives are offered. You'll become very aware of this advantage, especially when considering a country list with almost 200 countries.

Dropdowns are often used in the design of forms when requesting facts, for example, gender or age. This type of form is often found when shopping online or creating credentials for online services. Especially in the use case of forms where usually all the information must be filled in, you should pay special attention to crafting a neutral representation of the dropdown.

A *neutral display* indicates the state the dropdown assumes before the user makes an active choice. You can design this neutral state in two ways:

- You can make a pre-selection from the answer options.
- You can use a selection prompt.

If you can be relatively sure which selection the user will likely make, you can work with dropdowns by providing an answer option (*default*). For example, perhaps you need to collect contact data for a either a gynecology or urologist practice. In this case, a very high probability will respond on the form as either women or men, respectively. Thus, for default options, if you select "female" for the gynecology practice or "male" for the urologist practice, you can be reasonably sure that you've made a good preselection for the majority of users (and all others can still choose).

In all cases where you're less sure about the answer, we recommend you display a selection prompt, as shown in Figure 11.21, on the right. Note, however, that prefilled dropdowns can also have disadvantages. We'll talk about these disadvantages in a moment.

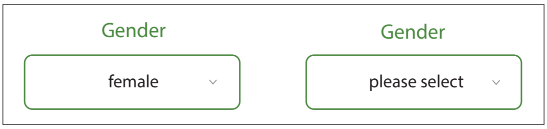


Figure 11.21 Dropdown with different neutral positions. A preselection is made on the left, and a prompt is shown on the right.

In case you present a selection prompt, as shown in Figure 11.21, on the right, there are again two possibilities:

- This field is a required field.
- This field is an optional field.

If answering the question to which the dropdown belongs is mandatory, usually after the first click on the dropdown, the selection prompt is removed, leaving only the content options, here "male," "female," "prefer not to say," etc., available. If the response is optional, either the "please select" response option remains or another option is added, which indicates that the user did not select an answer.

This pattern ensures that you only have real answers in your data, as pre-filled dropdowns are often overlooked. To make things even more complicated, pre-filled dropdowns also don't generate a warning since a selection has already been made (if you do not account for this issue technically).

Another point to consider when using dropdowns is the *length of the answer options*. Since a typical dropdown field usually has space for one or at most two words, long answers (e.g., whole sentences) can be more challenging to display with dropdowns; otherwise, the answer options will be truncated in the display or jump to the next line. Avoid truncating text at all costs. Multiline text should also be avoided as much as possible, or at least make the user aware of this by clearly dividing the options visually.

If you're designing for smartphones, dropdowns present another challenge. Especially long lists can be difficult to use. Therefore, smartphone operating systems often display dropdowns as a scrollable list, as shown in Figure 11.22. Radio buttons, which we'll discuss in the next section, are more suitable for short lists.

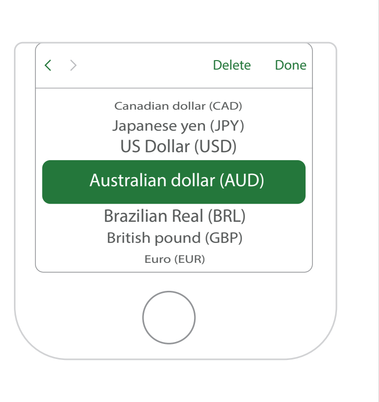


Figure 11.22 Displaying dropdowns on smartphones.

11.6.3 Radio Buttons

Radio buttons are the "little brothers" of dropdowns. You use radio buttons when the user should select only one of several options. In contrast to dropdowns, however, with radio buttons, the answers remain permanently visible, as shown in Figure 11.23.

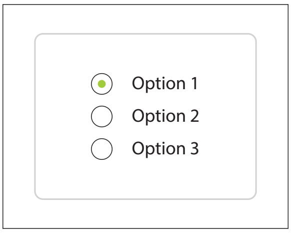


Figure 11.23 Typical radio buttons.

Because the answer options remain visible with radio buttons, radio buttons take up more space. On the other hand, the answer options can be much longer than with dropdowns since a selection field does not limit them.

Usually, radio buttons are used when changing one's own selection decision at any time should be easy, and the other options should be visible at any time. This strategy allows the user to compare options better than having a (long) dropdown list on the screen. Radio buttons are, therefore, particularly suitable when there is more than one correct answer (contrary to selecting a gender or a country when sending a package) but tangible options. For example, perhaps the user needs to choose a suitable payment method or choose between an in-person appointment or remote appointment. So, most of the time, with radio buttons, it's less about asking for facts and more about making adjustments according to a decision.

Radio buttons also significantly simplify the visual design of the response categories. For example, logos or images can be stored in addition to text to support users' decisions, as shown in Figure 11.24.

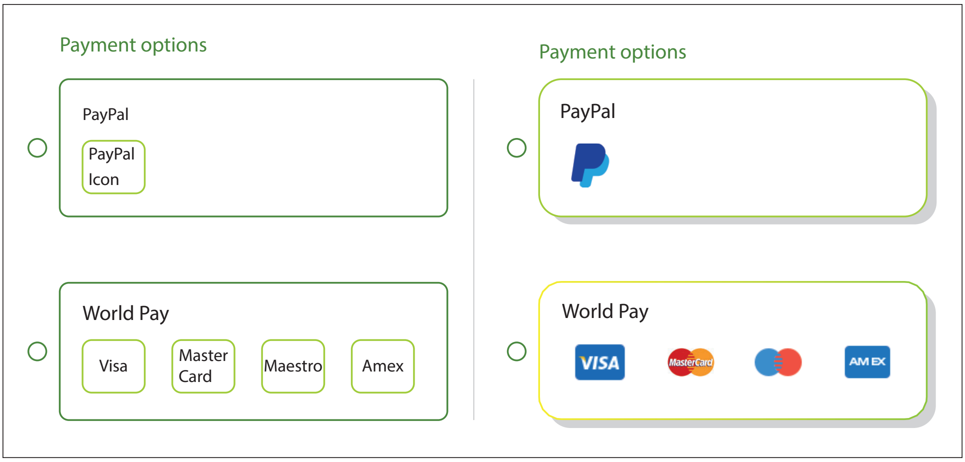


Figure 11.24 Selecting the payment method in an online store.

Offering radio buttons is particularly useful when it must be possible to change the decision spontaneously and/or only a few answer options exist. If you offer more than five answer options, other elements (e.g., dropdowns) are often used instead to provide an appealing visual presentation.

If you use radio buttons on smartphones, you can support usability by enlarging the touch areas. The circle alone is often too small an area to be pressed comfortably (unless you can make the circles larger). However, you can choose a button-like design instead of the typical circle design. The advantage of this tactic for smartphone users is that the touch areas are much larger, and thus usability is increased, as shown in Figure 11.25.

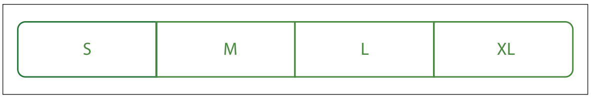


Figure 11.25 Alternative display of radio buttons particularly suitable for smartphones.

If you use the variant radio buttons shown in Figure 11.25, you should place the individual buttons directly next to each other without spacing to reduce the risk of confusion with buttons. In addition, you must use highlighting to distinguish the selected option from the others clearly.

11.6.4 Input Fields or Text Fields

Input fields or text fields often occur in digital interactions when the content of a field cannot be predefined but must be entered freely by the user. For example, this field is used in forms for entering first names, last names, or addresses, as shown in Figure 11.26.

In theory, creating a list of all first names occurring in a country would be possible. Practically, using such a list would make little sense since selecting from a list of over 2.7 million entries (yes, that's the actual number of first names registered in the US) would take an infinite amount of time and be rather error prone. Furthermore, the list would have to be kept up to date and would still present users whose first name is not included with problems.

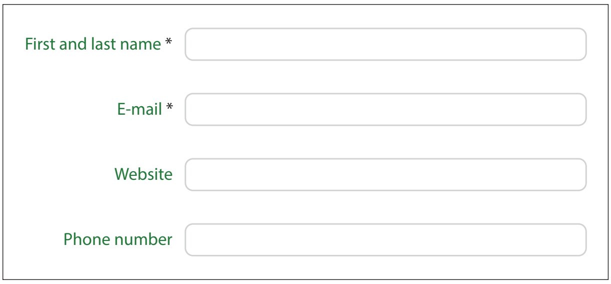
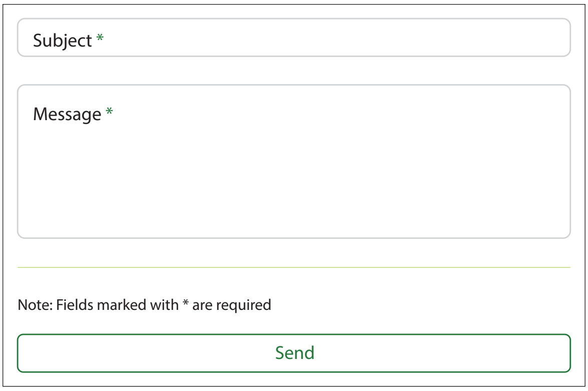


Figure 11.26 Input fields or text fields in a typical form.

Text fields are often formatted to fit the expected content. If only one or two words are required to be written, the text fields are usually short and on a single line. If longer text is expected, such as in a feedback form, multiple lines and a wider display are usually chosen, as shown in Figure 11.27.

One of the most common mistakes in the design of text fields and information display is when no distinction exists between them, and both look identical. Often, this problem arises when two empty white fields are displayed, only one of these fields already has something written in it by chance.

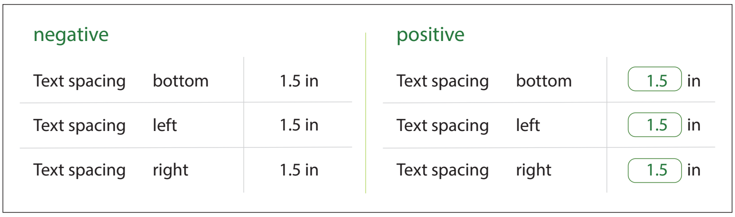


Note: Fields marked with * are required

Figure 11.27 Different sizes of text fields for different content. In the first field, only one line of text is expected, but in the second, several sentences or even paragraphs.

This confusion often leads to users adjusting the default texts, adding something, or deleting information. However, input or text fields are not intended for the (pure) output of information. Therefore, fields that allow a user to enter information must be designed differently, and this design must be exclusive to these fields.

Suppose you want to distinguish text fields requesting input from information displays. In that case, we strongly advise not drawing a border around the information, which would make them look like an input field. The same happens if you change the background color around the information compared to the page background. On the other hand, you should use a border for input fields (e.g., text or search fields). Additionally, you can support the distinction easier using another (typically lighter) background for input fields than the rest of the page, as shown in Figure 11.28.



negative

Text spacing	bottom	1.5 in
Text spacing	left	1.5 in
Text spacing	right	1.5 in

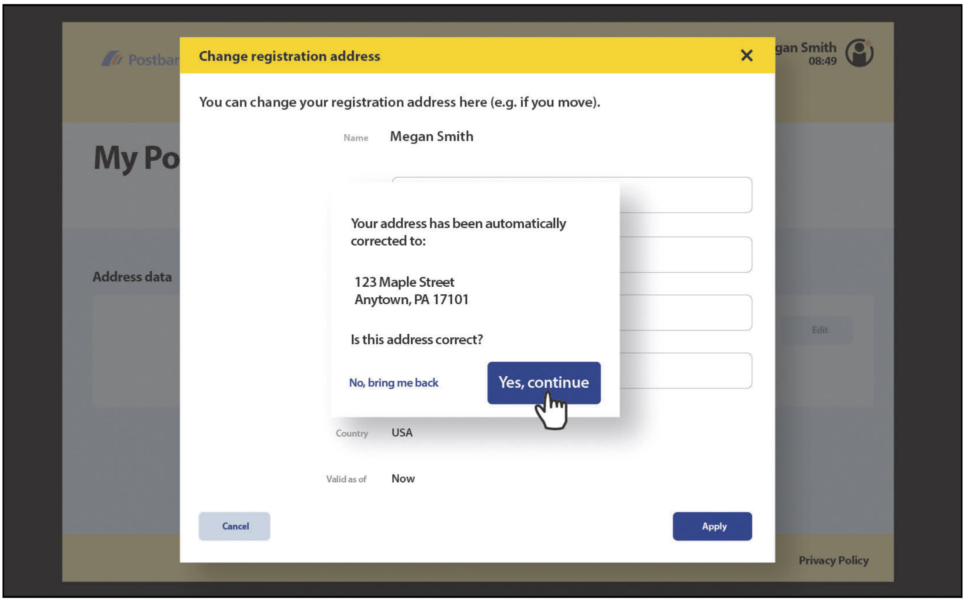
positive

Text spacing	bottom	1.5	in
Text spacing	left	1.5	in
Text spacing	right	1.5	in

Figure 11.28 Suppose the digit fields are input fields. Then the left is a negative example, and the right is a positive example. On the left, you cannot distinguish output from input fields. On the right, you can.

Text fields are very flexible in their use but also have the disadvantage that the quality of the entered data is difficult to verify. For example, a spelling error in a name field is technically challenging to check. In some cases, such a spelling error is just annoying. But if entry into a country is denied based on a spelling error, serious consequences can arise. Therefore, if you use text fields, you should also consider the data quality.

Text fields are also often used for address data. If users make mistakes in this context, online orders could be delivered to the incorrect addresses, for example. In this case, the system occasionally works with automated correction, automatically correcting the suspected typing errors or offering the user suggestions for improvement, as shown in Figure 11.29.



Change registration address ×

You can change your registration address here (e.g. if you move).

Name Megan Smith

Your address has been automatically corrected to:

123 Maple Street
Anytown, PA 17101

Is this address correct?

No, bring me back | Yes, continue

Country USA

Valid as of Now

Cancel | Apply

Figure 11.29 Correction of the entered address, including query whether the correction matches the user request (translated from https://www.postbank.de).

A similar approach can also be found in *search fields*, which look similar to text fields and are used similarly. In a search field, it is also usually impossible to know in advance what text the user will enter, which is why an empty field often represents a search field. To make this field more obviously a search function, search fields are regularly combined with magnifying glass icons or a “search” text prompt, as shown in Figure 11.30.

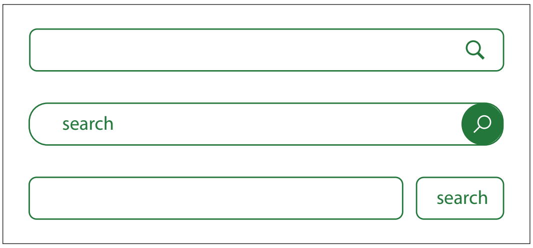


Figure 11.30 Different representations of a search field as a particular input field.

You may be wondering why the probably best-known search field in the world—namely, that of *Google*—does not appear here as an example. *Google Search* is not a classic text field or a search field at all, but a further advancement of the classic search field.

Unlike conventional search fields, you can not only enter text and search for it, but you'll receive suggestions for search terms based on your search, which you can then select with one click and make the search request, as shown in Figure 11.31. Thus, Google Search offers further advantages compared to standard search fields since you're also offered assistance finding suitable terms. This search box form also leads us to another interface element that works similarly but is not the same: the combo box.

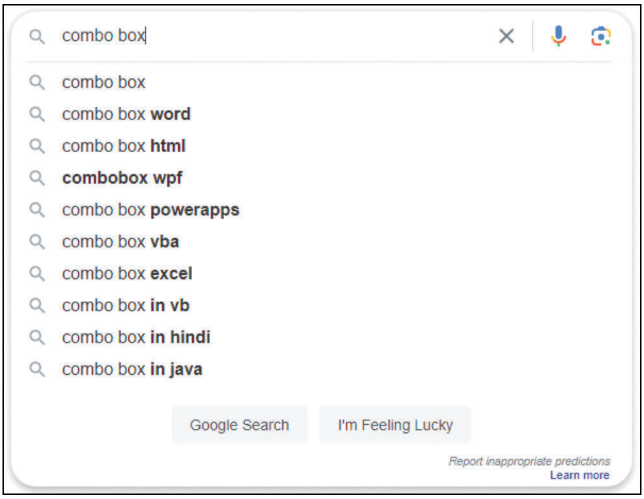

combo box
combo box
combo box **word**
combo box **html**
combobox wpf
combo box **powerapps**
combo box **vba**
combo box **excel**
combo box **in vb**
combo box **in hindi**
combo box **in java**
Google Search
I'm Feeling Lucky
Report inappropriate predictions
Learn more

Figure 11.31 The Google Search box.

11.6.5 Combo Boxes

Combo boxes represent—as the name suggests—a combination of two different input methods. On the one hand, the combo box is a text field in which free text can be written. On the other hand, it is a kind of dropdown where a relevant result can be selected from the entries by means of a click, as shown in Figure 11.32.

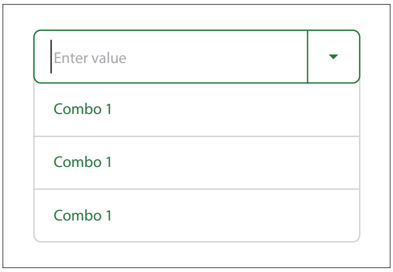

Enter value
Combo 1
Combo 1
Combo 1

Figure 11.32 Typical combo box.

Using combo boxes is especially helpful when choosing from many different answer options to get to the desired answer quickly. Users start writing in the text field and immediately get to the corresponding place in the list of answer options. A typical application for combo boxes is the selection of a language since apps often offer more than 50 different languages or the selection of a country.

There are different variants for the implementation of combo boxes. Either only the first letter can be entered, and the entry of each further letter leads to a new search again. Or you can enter several letters or words as in a "real" search box. To illustrate the difference: Imagine you would like to select the country "Canada" as your place of residence from the list. In the first case, you would type "C," and then you would get to the list of all countries starting with "C." If you now type "a," the list will not be further reduced to countries starting with "Ca," but you would jump in the list to countries starting with "a," for example, Australia or Argentina. You can probably imagine which implementation is better from the user's point of view.

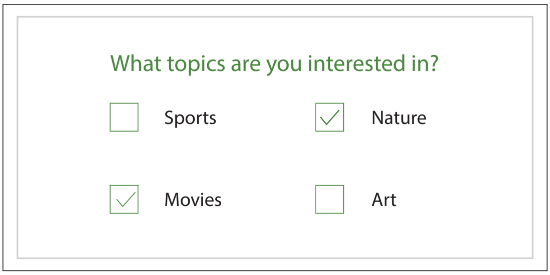
Compared to dropdowns, combo boxes have an advantage in that they allow a significantly longer list of choices since search is supported. The advantage that little space is used also remains. Unfortunately, the disadvantage remains the same: The description of the options must usually be relatively short. Otherwise, the end will be truncated or double-spaced by the box.

Especially in combo boxes, we regularly see designs that do not take this principle into account, and the reason for this omission is the desire for good usability. In general, the creators of combo boxes try to make the selection easier for the users by creating categories of answer options and then writing the details afterward. Unfortunately, the details could be cut off, making a meaningful selection impossible. So, *be careful with the option descriptions!*

This case is especially true on small displays such as smartphones. Combo boxes can be a real challenge for users in this context because often, combining typing (in that case, the keyboard folds out) and scrolling (the keyboard folds back in) leads to poor usability. Truncating text and multiline text can also become problematic, as you have much less space. However, combo boxes are not completely unsuitable for smartphones. Especially for long lists, combo boxes also offer great advantages over dropdowns.

11.6.6 Checkboxes

Another interaction element is a *checkbox*. Checkboxes are used when not only one option, but several options can be selected simultaneously from several alternatives. The boxes are usually empty in their initial state and are selected by clicking on them. This box is deselected again by clicking again on a box that has already been selected, as shown in Figure 11.33.



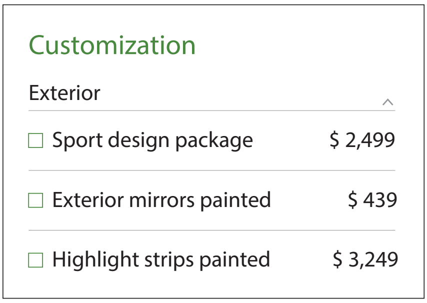
What topics are you interested in?

☐ Sports ☑ Nature

☑ Movies ☐ Art

Figure 11.33 Typical checkboxes.

In contrast to the alternatives we've presented so far, checkboxes can significantly shorten interactions since only one overall query needs to be performed instead of many individual queries. For example, checkboxes are often used in configurators where multiple options can be selected simultaneously, as shown in Figure 11.34.



Customization

Exterior

☐ Sport design package	$ 2,499
☐ Exterior mirrors painted	$ 439
☐ Highlight strips painted	$ 3,249

Figure 11.34 Using checkboxes in a configurator to select extra features.

Checkboxes are a straightforward interaction option. However, care must be taken during implementation to maintain clarity. For example, in configurators, this clarity can be achieved by placing visual separators between the individual options or by breaking down the decision into individual categories (in our example, **Exterior**). In this way, several selection groups of checkboxes can be provided.

Another critical point is the *size* of the checkboxes and the *distance* between them. Especially on mobile devices, you should ensure the control surface is large enough to hit a checkbox safely. Also, the boxes should not be so close together that the incorrect box is checked by mistake.

An error often observed in the design of checkboxes is marking the checkboxes with an X instead of a checkmark. However, using X can lead to misunderstandings when viewing the checkboxes, as it can give the impression that a box is crossed out, as shown in Figure 11.35.

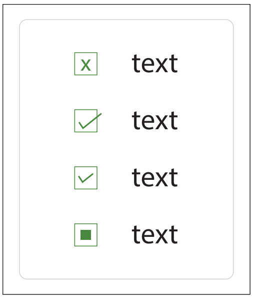
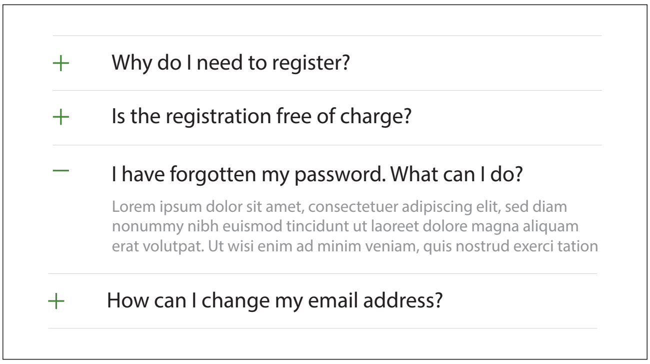


Figure 11.35 Checkbox variants.

11.6.7 Accordions

An accordion is an interactive element that can collapse and expand groups of elements, such as the checkboxes we just introduced, as shown in Figure 11.36.



+ Why do I need to register?

+ Is the registration free of charge?

− I have forgotten my password. What can I do?

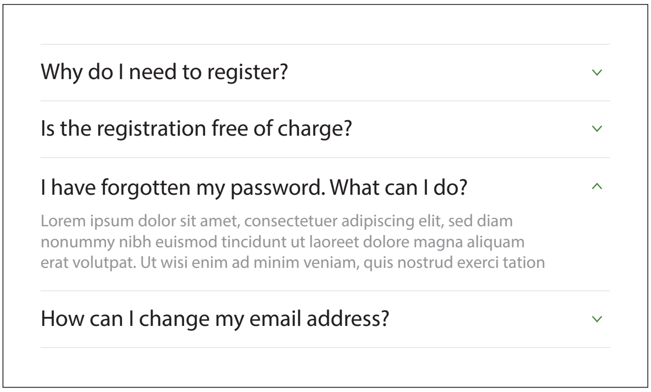
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation

+ How can I change my email address?

Figure 11.36 Typical accordion in a frequently asked questions (FAQ) area, where all questions are visible, but the answers can be unfolded.

Accordions are typically used when large amounts of information are available, not all of which is important simultaneously. This scenario could be the case, as in the previous section, with a configurator, but accordions are also often used in FAQ sections, so only the questions are visible, and the users must actively unfold the answer they desire.

Most accordions are designed so that the individual areas are unmistakably separated from each other, either by color or by lines. In addition, each area has its interaction element to expand the area. Usually, plus sign or downward arrow icons are used to expand an area, and minus sign and upward arrow icons are used to collapse it, for example, as shown in Figure 11.36 and Figure 11.37.



Why do I need to register? ⌄

Is the registration free of charge? ⌄

I have forgotten my password. What can I do? ⌃

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation

How can I change my email address? ⌄

Figure 11.37 Alternative design of the accordion with arrow signs.

In addition to the visual design variants, accordions come in different *versions* regarding interaction behavior. One design variant allows only one area to be opened at a time. Thus, whenever you open an area and another is already expanded, the previously already open area is automatically collapsed. By automatically collapsing the areas, the page remains clearer, and the focus is always clearly on the area that was just selected. Sometimes—if the previously open area is above the new question—the interaction feels a bit "jerky" because the view "jumps" due to the collapse. Since reading is mostly from top to bottom, this jumping can occur often.

In contrast, in another design variant of the accordion, any number of areas can be open simultaneously. Of course, this approach eliminates jumping, but the content on the page can also become rather confusing, especially FAQ pages, where the answers to questions can sometimes be quite detailed. When articles are expanded, the screen quickly turns into long scrolling paths.

11.6.8 Toggle Buttons

A relatively new control element is what's called a *toggle button*. A toggle button can be used to switch between different states. In most cases, two states are distinguished: "on" versus "off," as shown in Figure 11.38.

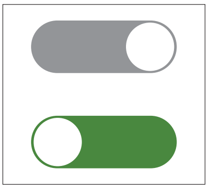


Figure 11.38 Typical toggle buttons.

The toggle button is a control element typically used on mobile devices. Switching from "on" to "off" or from "off" to "on" is done by swiping the button in the opposite direction. Pressing or touching the entire button also often triggers toggling. In many cases, the toggle button is color-coded so that a green color indicates the option is activated. A gray color means that the option is disabled.

Toggle buttons are mainly used for settings, such as activating airplane mode or switching on a hotspot. In principle, toggle buttons are always suitable when switching between "on" and "off" options is necessary.

Some toggle buttons are additionally labeled (remember the color vision issue) to increase uniqueness. The labeling can be either on the button itself or on the edges. Sometimes, icons are also used, as shown in Figure 11.39.

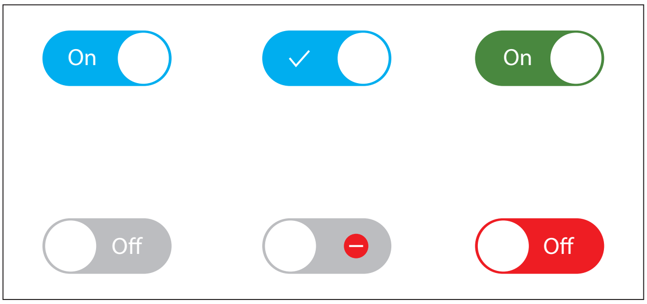


Figure 11.39 Various toggle buttons also use icons and labels to increase uniqueness.

Toggle buttons are small and often quite aesthetic, which is why they are often used in interfaces. However, for toggle buttons to develop their full potential, their labels and states must be chosen clearly and sensibly. Sometimes, this clarity is a challenge, especially if you don't use the common representation of "colorless" for "deactivated" and "green" for "activated." With a single label, whether the label is intended to describe the possible future state of the button or the state of the button at this point can be unclear, as shown in Figure 11.40.

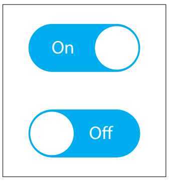


Figure 11.40 Which button is on and which is off?

Theoretically, toggle buttons can display more than just "on/off." In this case, a third stage is usually added to the center of the button, as shown in Figure 11.41. However, this interaction is relatively uncommon and is rarely used. The main disadvantage of this variant is that the middle option is not as easy to select (especially in mobile applications). A better option would be to consider radio buttons (or the mobile variants of radio buttons) instead of this option.

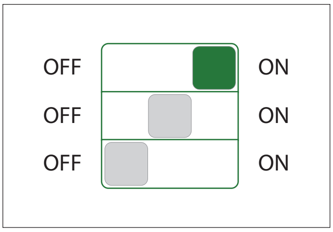


Figure 11.41 Toggle button with three options.

11.6.9 Sliders

While you can toggle between a few discrete states with a toggle button, you can set a continuous numeric value with a slider, as shown in Figure 11.42.

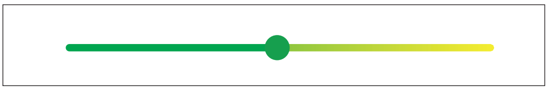


Figure 11.42 A typical slider.

Sliders are often used when it is essential to set a value approximately, but not exactly. Usually, a combination involving an input field is used for setting exact values.

Sliders can also define *value ranges* when two control surfaces are integrated, as shown in Figure 11.43. However, this feature is often more difficult to use, especially when designing for a device with touch controls.

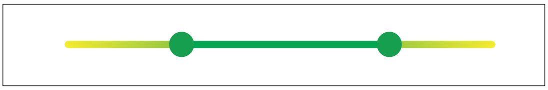


Figure 11.43 Slider for setting a value range.

A typical application for a slider is a loan calculator at a bank. In this case, numerical values like the terms, loan amounts, and interest rates can often be entered via sliders. Also typical for this application is that the sliders cannot be moved independently of each other. For example, the other values and their sliders also change if you adjust the terms through its slider.

Another typical use case for a slider setting a range of values is to select a price range during online shopping.

Though not as common, another use of a slider is to set a volume as a dial, as shown in Figure 11.44. (The more conventional linear slider, however, is more common than a slider as a dial.)

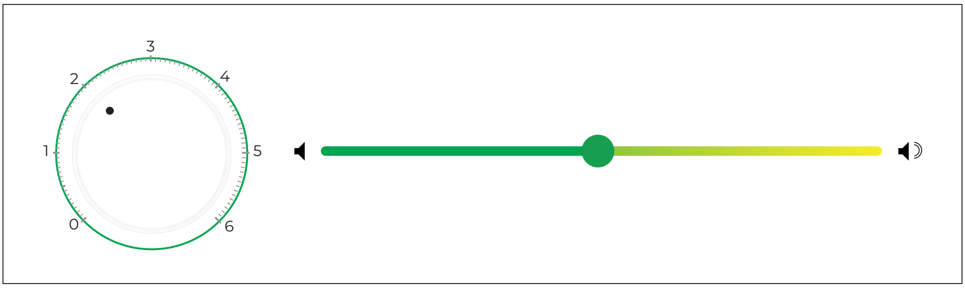


Figure 11.44 Slider for adjusting the volume (on the left as a rarer dial variant, on the right horizontally).

Unless you have a good reason for it, you should do without the dial variant. Usually, a dial is more difficult to operate than a horizontal (or vertical) slider. A dial variant will be imaginable if you have a parallel physical control element designed as a rotary knob that controls the same value.

11.6.10 Calendars

The selection of dates or times often occurs in the digital world when specifying birthdays or selecting appointments. In this context, input formats have also become established and are used almost universally.

In the past, dates and times were often entered freely. However, as different date and time formats must be brought together in an increasingly international world, free entry has proven to be increasingly prone to errors. To reduce this tendency toward error, different tactics are possible. The simplest way is to specify the required formatting, as shown in Figure 11.45.

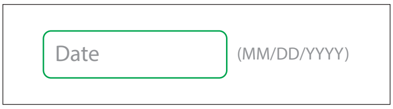


Figure 11.45 Input field for a date with a format specification.

By specifying formatting, the user is informed of which input type is expected before the input, increasing the probability that a correct entry will be made directly. Nevertheless, since the field remains a free field, incorrect entries can, of course, occur. Still, these entries can usually be corrected by displaying an error message with the required format afterward.

An alternative to this approach would be to use three separate fields for entering the day, month, and year, as shown in Figure 11.46. In this case, too, incorrect entries could occur. For example, a “44” could be entered as the day. However, input errors within the realm of possibility are more critical and, therefore, cannot be technically caught. For example, you could swap the day and month when entering 12/01/2023 without it being a technical error.

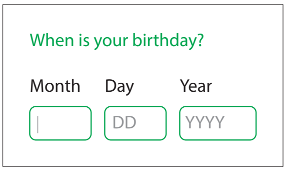


Figure 11.46 Entering a date via individual input fields.

To ensure that no incorrect selection is made in the first place, other ways to select a date exist.

A widespread variant is a calendar display, where a date can be selected with a click, as shown in Figure 11.47.

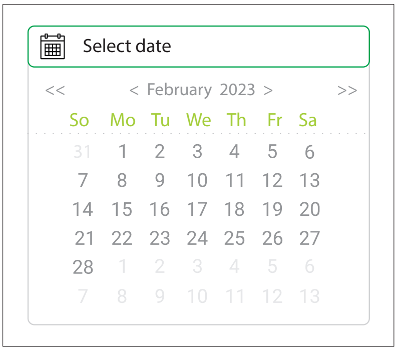


Figure 11.47 Selecting a date via a calendar.

The calendar element comes in different *variants*. In some implementations, the calendar has a button to confirm the selected date. In other cases, the date is automatically applied after the click. Besides displaying the days of the week and the date, calendar elements usually have a month selection in the header. This selection can usually be toggled back and forth with buttons, as shown in Figure 11.48.

Calendar elements are convenient when it comes to selecting dates that are in close proximity to the current time like the dates for special events or travel bookings, for example. They offer the advantage that the date is selected by click, and the input format is fixed, meaning no incorrect entries can be made except for selecting the wrong date. In addition, the calendar display usually provides users with further information to assist them in their selection (e.g., days of the week, holidays, etc.). After selection, the selected date is usually displayed again as text so that users can check it, as shown in Figure 11.49.

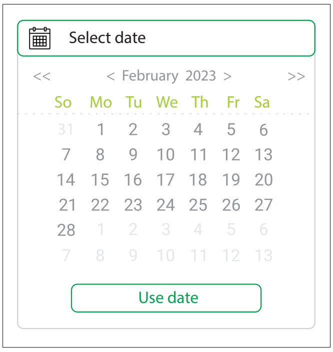

Select date
<< < February 2023 > >>
So Mo Tu We Th Fr Sa
31 1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 1 2 3 4 5 6
7 8 9 10 11 12 13
Use date

Figure 11.48 Alternative calendar display with a button to confirm the selection.

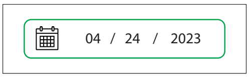


Figure 11.49 This date field starts with opening the calendar item and then shows the selection as text.

Using calendars for date selection is more complicated when the selected date is far in the past or far in the future. For example, if you want to select a person's year of birth, the person may have to go 50 years or more into the past. If you only offer the option to click back a month at a time, any user would have been faster with a free entry field. Thus, some calendar elements use additional options to speed up date selection. *Double arrows* are often used for this purpose, taking the person to the same month one year earlier or later, as shown in Figure 11.50.

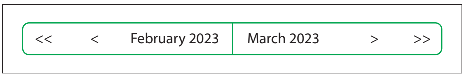

<<
<
February 2023
March 2023
>
>>

Figure 11.50 Clicking the double arrows jumps an entire year into the future or into the past.

Another variant is to offer an explicit *year picker*, which works like a dropdown and is part of the calendar element, as shown in Figure 11.51.

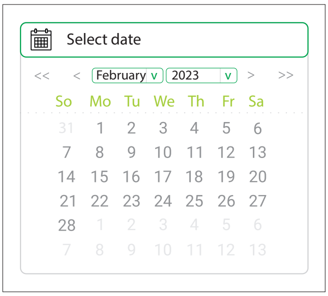

Select date
<< < February v 2023 v > >>
So Mo Tu We Th Fr Sa
31 1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 1 2 3 4 5 6
7 8 9 10 11 12 13

Figure 11.51 Calendar element that additionally allows a separate selection of the year as a dropdown.

As you can see, a calendar is composed of several different interactive elements. However, since it often appears as an overall element, we have decided to present it here in a single section.

A final alternative, which can be useful especially on mobile devices and touch-based displays, is called a *date picker*, where the data is displayed on a kind of wheel, as shown in Figure 11.52. Users then select the date by sliding the wheel up and down, thus changing the date. The interaction is simple and quick to understand and has become popular on many mobile devices. For non-touch displays, this selection variant is only suitable to a limited extent since pushing and pulling with the mouse is not a natural or learned movement pattern.

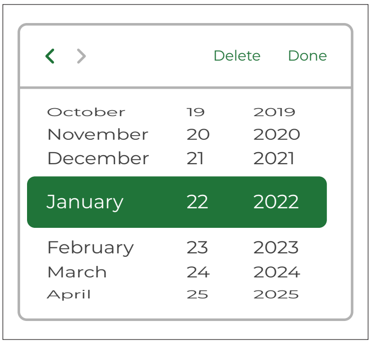


Figure 11.52 Date selection on smartphones.

The advantage of this type of date selection is that even large time differences, such as 20 or more years, can be bridged quickly by turning the wheel quickly. However, accuracy is reduced, so it is sometimes necessary to turn back and forth to set the exact date.

11.6.11 Time Pickers

As with the calendar elements, the input of times can be enabled via one or more input fields, as shown in Figure 11.53. In this case, too, input errors should be reduced by explaining the desired format.

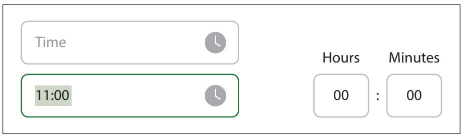
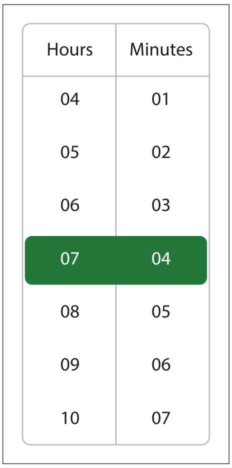


Figure 11.53 Different input fields for time.

Since the same problems arise in this context as with the date selection, various alternatives have been established regarding how a time can be entered by selecting predefined values.

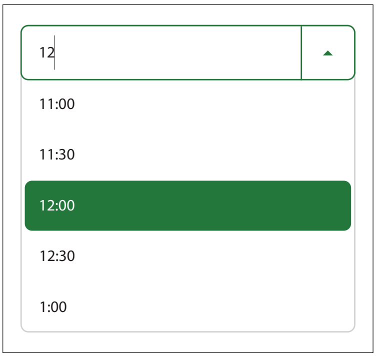
A common variant that also works well on smartphones is a wheel selection element, as shown in Figure 11.54.



Hours	Minutes
04	01
05	02
06	03
07	04
08	05
09	06
10	07

Figure 11.54 Selecting a time via a wheel selection.

Alternatively, dropdowns are occasionally used, especially when working with the 12-hour system, where the number of alternatives is manageable. For many computer applications, selecting the time by clicking as well as changing it in the text field is often possible, as shown in Figure 11.55. This variant makes it possible that often-used values are displayed as defaults in the dropdown (e.g., "0.5 h" for an appointment), and still, a more precise entry is possible.



12

Figure 11.55 Selection of a time via a dropdown with often used times as well as the possibility to write directly into the field.

In mobile variants, *Android Material Design* also specifies a separate element that selects the time based on a clock, as shown in Figure 11.56. The hour hand is set first, followed by the minute hand.

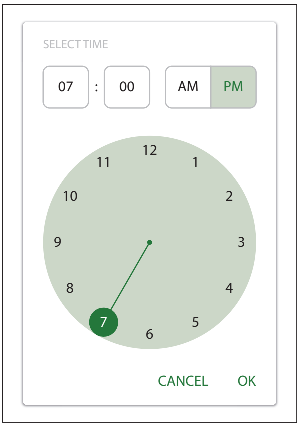
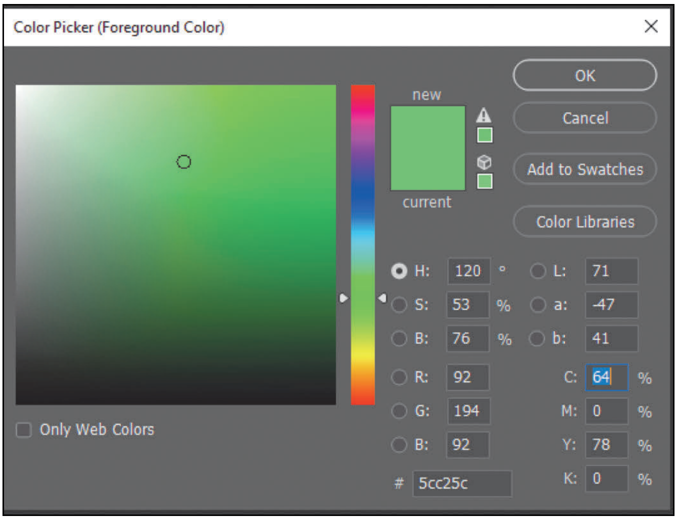

SELECT TIME
07
:
00
AM
PM
12
1
2
3
4
5
6
7
8
9
10
11
CANCEL
OK

Figure 11.56 Android Material Design's time selection clock.

11.6.12 Color Selection

A special element called a *color picker* is used when a specific color must be selected (e.g., for a font, for a shape) that is not yet predefined and thus cannot be selected via a button, as shown in Figure 11.57.


Color Picker (Foreground Color)
OK
Cancel
Add to Swatches
Color Libraries
new
current
Only Web Colors

● H: 120 °
○ S: 53 %
○ B: 76 %
○ R: 92
○ G: 194
○ B: 92
○ L: 71
○ a: -47
○ b: 41
C: 64 %
M: 0 %
Y: 78 %
K: 0 %
5cc25c
☐ Only Web Colors

Figure 11.57 Color selection in Adobe Photoshop 2023.

Usually, with color pickers (as shown in Figure 11.57) there are different ways to specify a color. For example, the user could directly input a hexadecimal color code or select the desired color via a color wheel or slider.

The selection is usually performed in two steps. First, a color bar or color wheel is used to select the primary color, for example, green. Then, by moving the cursor in the color field, the color's intensity is determined by adding more or less white or black.

Since comparing colors when not displayed next to each other is often challenging, many color picker elements also contain a before-after comparison, which compares the original color with the new one, as shown in Figure 11.58.

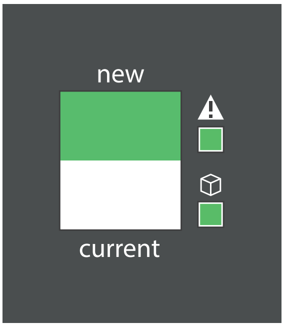


Figure 11.58 Adobe Photoshop 2023 color selection element. The newly selected color (green) is displayed next to the previous color (white).

This feature allows users to adjust the color until the desired hue is achieved. Afterward, the new color is usually applied by clicking **Apply** or **OK**. In most cases, the applied color is also provided as a quick selection to apply the same color tone quickly.

Also widely used are *color palettes* that allow quick color selection, as shown in Figure 11.59. Color palettes can also be predefined if the exact color scheme is unimportant.

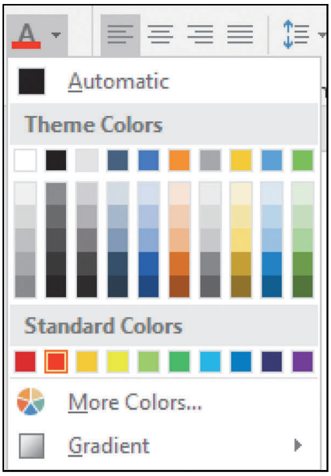
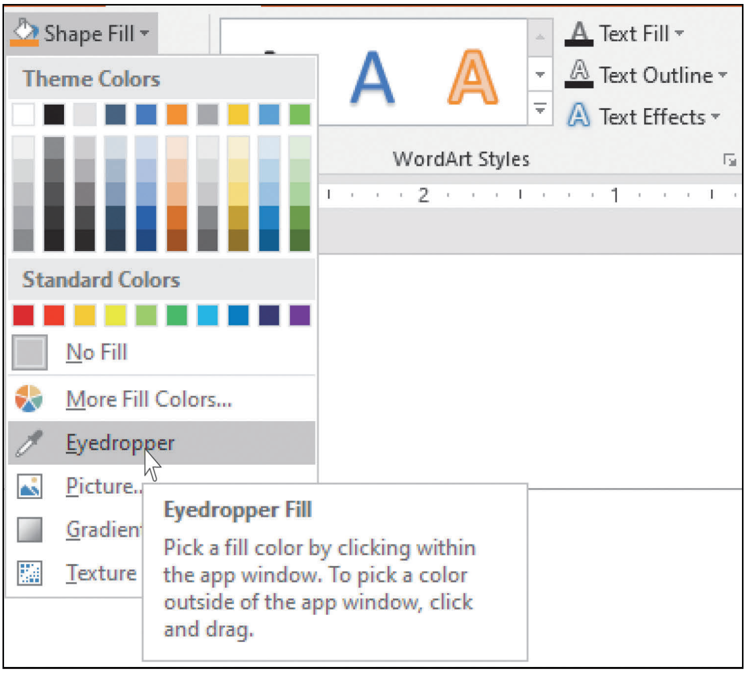


Figure 11.59 Selecting a color using predefined color palettes in Microsoft Word 2019.

Another convenient and straightforward variant for the selection of colors is the *eyedropper*, as shown in Figure 11.60.

With the eyedropper, you can pick up and use any existing color. For many use cases, this capability is extremely helpful since, when using colors, care is usually required to use the same colors and/or matching colors.


Shape Fill
Theme Colors
Standard Colors
No Fill
More Fill Colors...
Eyedropper
Picture...
Gradient
Texture
Text Fill
Text Outline
Text Effects
WordArt Styles
2
1

Eyedropper Fill

Pick a fill color by clicking within the app window. To pick a color outside of the app window, click and drag.

Figure 11.60 The eyedropper tool in Microsoft PowerPoint 2019.

11.6.13 Information Buttons, Info Buttons, and Tool Tips

Information buttons are a special type of button. You may also know them as "info-i" or "info-?" somewhere near text or controls, as shown in Figure 11.61. These *information buttons* are clickable and show you additional information if needed.

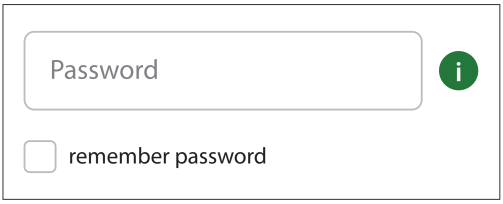
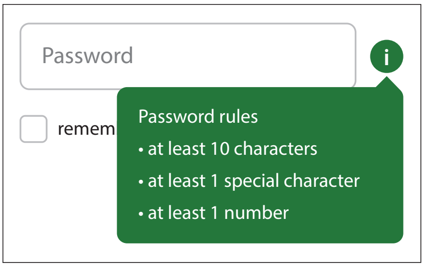

Password
i
☐ remember password

Figure 11.61 Information button at the edge of an input field.

Mostly this display of information happens in the form of an additional opening popup (see also Section 11.6.14) that overlays the current screen and is often placed near the info icon. This type of popup is also called a *tool tip*, as shown in Figure 11.62.

In some cases, however, text appears below the info icon, providing further information, as shown in Figure 11.63.



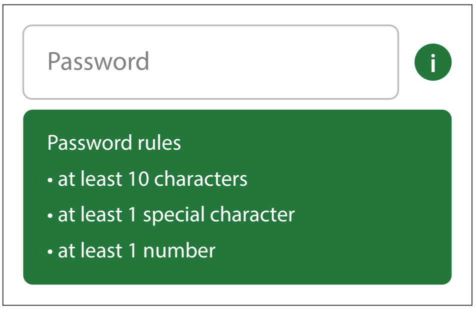
Password

☐ remem

Password rules

- at least 10 characters
- at least 1 special character
- at least 1 number

Figure 11.62 Tool tip providing additional assistance to the user when needed.



Password

Password rules

- at least 10 characters
- at least 1 special character
- at least 1 number

Figure 11.63 Explanatory text that expands like an accordion when you click on the info icon.

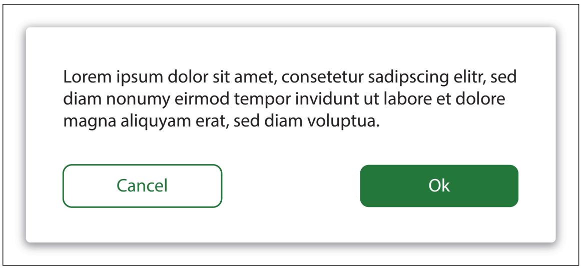
Information buttons are always helpful when providing specific information about a particular interface element. For example, in *forms*, you'll want to explain to users what the requested information is needed for or what information is needed. Similarly, with *password fields*, you'll want to explain the password requirements.

Information buttons are practical to implement, as it becomes possible to work locally and keep the structure simple. Nevertheless, you should think carefully about how many information fields you need. By integrating them into other elements or by placing them directly next to each other, a screen with many information fields, unfortunately, looks crowded and less tidy.

11.6.14 Popups

Popups are a classic interface element that has been used for a long time, as shown in Figure 11.64. Popups interrupt a user's current workflow to display information that users must respond to.

Popups are usually triggered by a user action. For example, triggers could include submitting a form or approaching the browser bar at an online store, which triggers the "Are you sure you want to leave this page?" popup. But some popups arise that a user does not trigger. Error messages, for example, can be displayed even without a specific user action and are often presented as popups.



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

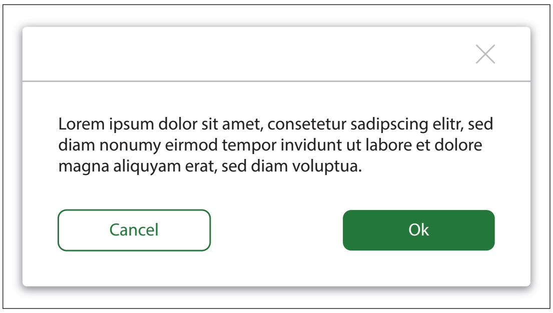
Figure 11.64 Typical popup window that "overlays" the actual UI and draws the focus of interaction to itself.

What all popups have in common is that, after being triggered, the popup covers the screen—usually in the center—and interrupts further interaction with the interface. The popup only disappears when the user performs another action, for instance, confirming the popup dialog or clicking away from the popup.

A typical application field for popups is a *confirmation dialog* in which a user must release a performed action once again before it can be executed. This type of dialog is often used, for example, when deleting or releasing data. But popups are also used to stop the current interaction when users display undesirable behavior from the operator's point of view, for instance, leaving a page or exiting an application, among others.

In our view, popups are justified if they prevent harm to users—as would be the case if data were deleted unintentionally. However, we find that the use of popups is not appropriate from the users' point of view if purely in the monetary interests of the provider. The "Are you sure you want to leave?" popups on blogs often equate to a poorer UX if they are designed to be intrusive.

When implementing popups, several variants are available. In the classic variant, the popup can be closed only by clicking one of the buttons (usually these popups have one button to confirm and one button to cancel) or by clicking an X in the upper-right corner, as shown in Figure 11.65. This classic popup is usually centered over the existing screen.

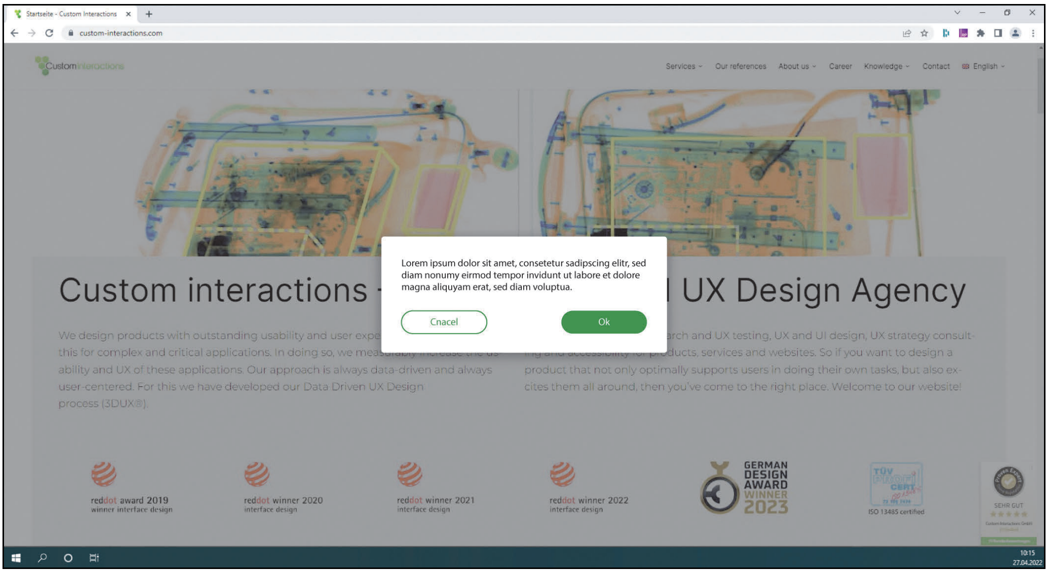


Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Figure 11.65 Classic popup with three buttons: Cancel, Confirm, and X, which also functions like the Cancel button.

What's called an *overlay* is usually used in more modern popups. This popup is displayed in the center of the screen, but the rest of the screen is darkened at the same time, as shown in Figure 11.66. This variant places the viewing focus on the popup since it stands out more clearly from the background. In addition, the overlay offers an additional variant for closing the popup quickly and in a user-friendly way since clicking in the darkened area is usually sufficient for closing the popup.

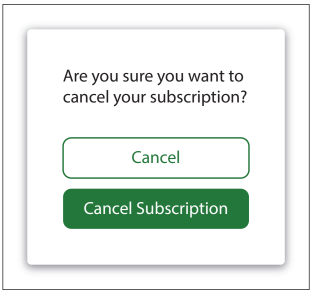
A common mistake when using popups is choosing the wrong button labels in the dialog. In English, one occasionally encounters the double meaning of the word "cancel," as shown in Figure 11.67.



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Cnacel
Ok
Startseite - Custom Interactions
custom-interactions.com
Services
Our references
About us
Career
Knowledge
Contact
English
Custom interactions
UX Design Agency
reddot award 2019 winner interface design
reddot winner 2020 interface design
reddot winner 2021 interface design
reddot winner 2022 interface design
GERMAN DESIGN AWARD WINNER 2023
ISO 13485 certified
10:15
27.04.2022

Figure 11.66 A popup as an overlay. Note the typical feature: the remaining part of the UI is darkened.



Are you sure you want to cancel your subscription?

Cancel
Cancel Subscription

Figure 11.67 Misleading buttons due to the double meaning of "cancel."
It would be better if the Cancel button were labeled "Keep Subscription."

This vagueness occurs especially when the same answers are to be used for different buttons. In this sense, a bit of effort might be needed to come up with labels that are specific enough and still work. From our point of view, the best version is the following: Usually, the dialog asks a question, such as "Are you sure you want to delete the data?" The buttons should then be appropriate to answer the question from the user's perspective—a simple "yes" or "no." But even better would be "Yes, delete" or "No, cancel," as this tactic makes it unmistakably clear what action is triggered by the click. Avoid double negations like "No, do not delete." The "do not" may be overlooked during a quick skim, and only the "delete" is seen—which forces users to start the dialog again.

In general, you should use popups sparingly. Since popups constantly interrupt the current activity and make the process longer and slower, a crucial task is to consider carefully beforehand when using a popup is justified. As a rule of thumb, you should always use popups when the action will cause tremendous or unrecoverable harm to the user.

11.6.15 Menu Bars

Many applications have some form of menu bar. Usually located at the top of an application or web page, menu bars contain cross-references to other content, as shown in Figure 11.68.

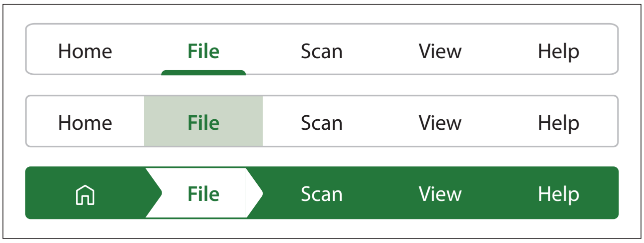


Figure 11.68 Different menu bars.

A helpful tactic is to have a menu whenever a lot of information cannot all be displayed simultaneously. A menu allows your users to switch between different content. Additionally, the menu allows you to prioritize content. The most important content usually comes first, and the less important a piece of information is, the further back it goes. Menu bars exist horizontally and vertically, with the horizontal orientation becoming quite popular, especially in the web context. The popularity of this orientation is evidenced by the fact that nearly all major brands use a horizontal orientation on their websites, as shown in Figure 11.69.

Exceptions are websites that were primarily developed for mobile devices such as smartphones. In this case, you'll often find a vertical menu folded up in its original state, which only opens when you click down. In this case, you'll often find a vertical menu folded up in its original state, which only opens when you click, as shown in Figure 11.70. This is often called a burger menu because the icon for the menu, with its three layers, resembles a hamburger.

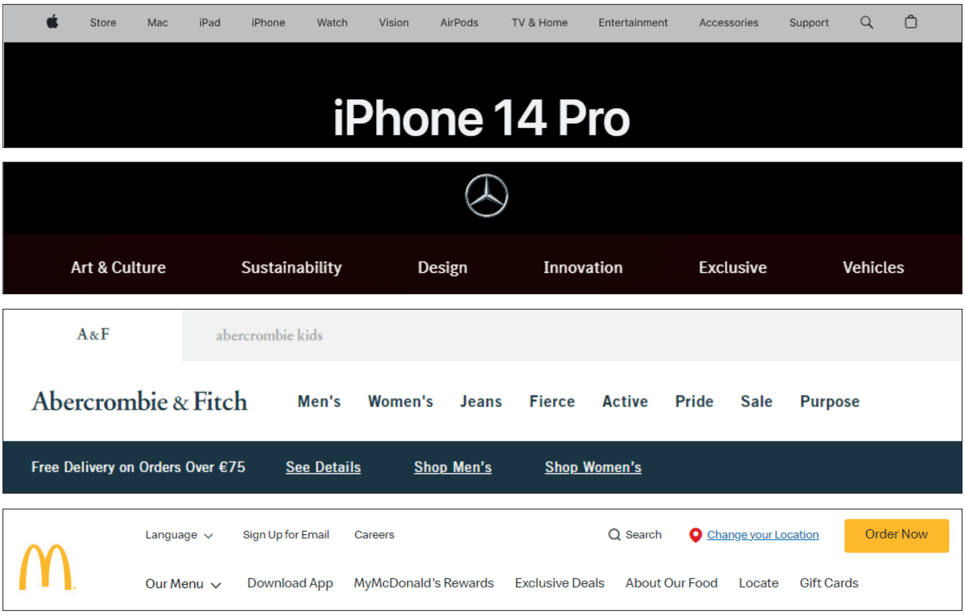

Store
Mac
iPad
iPhone
Watch
Vision
AirPods
TV & Home
Entertainment
Accessories
Support
iPhone 14 Pro
Art & Culture
Sustainability
Design
Innovation
Exclusive
Vehicles
A&F
abercrombie kids
Abercrombie & Fitch
Men's
Women's
Jeans
Fierce
Active
Pride
Sale
Purpose
Free Delivery on Orders Over €75
See Details
Shop Men's
Shop Women's
Language
Sign Up for Email
Careers
Search
Change your Location
Order Now
Our Menu
Download App
MyMcDonald's Rewards
Exclusive Deals
About Our Food
Locate
Gift Cards

Figure 11.69 Horizontal menu bars at www.apple.com, www.mercedes-benz.com, www.abercrombie.com, and www.mcdonalds.com.

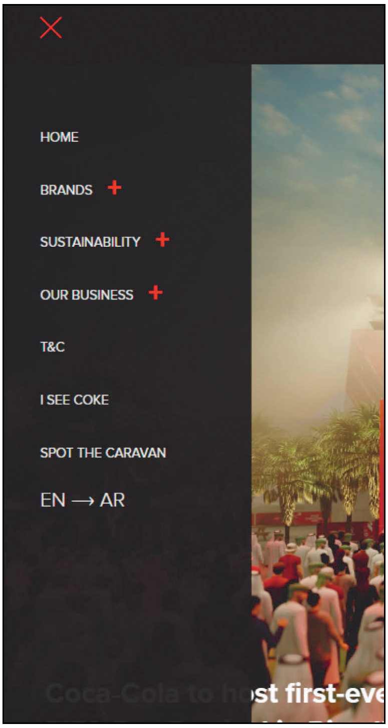

HOME
BRANDS +
SUSTAINABILITY +
OUR BUSINESS +
T&C
I SEE COKE
SPOT THE CARAVAN
EN ⟶ AR
Coca-Cola to host first-eve

Figure 11.70 The hamburger icon opens a vertical menu bar on the Coca-Cola Middle East webpage (https://en.coca-colaarabia.com).

Which menu you prefer should depend on the device on which the content is displayed and which content exactly is shown. A best practice is to also consider whether the *operation* is likely to be *touch-based* or *mouse-based*. If the operation is touch-based, you should also think about issues such as occlusion because with touch operations, you always reach across the screen, and content becomes physically invisible.

A typical mistake we've seen in menu design is prioritizing content from the company's point of view rather than from the user's point of view. (You can counter this problem with the *card sorting* method, which we present in detail in Chapter 12, Section 12.17.) For example, business departments that are utterly incomprehensible to outsiders may still be presented, but including these departments is not helpful to users.

This problem is especially evident on websites where the first content is "About Us" or "About [Company Name]." We know that statement is probably painful and will make you gulp at first, but in truth, starting with an "About" page is not advisable because of the following:

- Only people who want something from you are interested in the company (i.e., job applicants or salespeople).
- All the people you want something from (i.e., potential customers) are interested in your offers but are often not interested in your company itself.

The only exceptions to this rule are potential investors and when you sell expert services. But suppose you want to use your website to provide information for potential customers. In this case, the first items on your menu should also be about your potential customer's questions, tasks, and wishes. In hindsight, you should thus start, for example, with the store, the presentation of the products, case studies, or testimonials from customers and only later work on the other aspects.

If you're designing an application and not a web presence, start with the tasks that are run repeatedly and regularly, and arrange infrequent tasks further down the menu.

11.6.16 Carousels

A carousel displays several elements next to each other, through which the user can move step by step or by scrolling, as shown in Figure 11.71.

Carousels are mainly used when graphical information is to be "browsed." This representation is used, for example, by Apple in an element it called the *cover flow* for music album covers and images, allowing users to interact playfully while browsing the collection, as shown in Figure 11.72.

A helpful tactic is to contain core information in the image for a carousel to make sense. Theoretically, texts can also be displayed in this way—quotes, for example—but the great advantage of this display method is that images can be skimmed more quickly than texts.

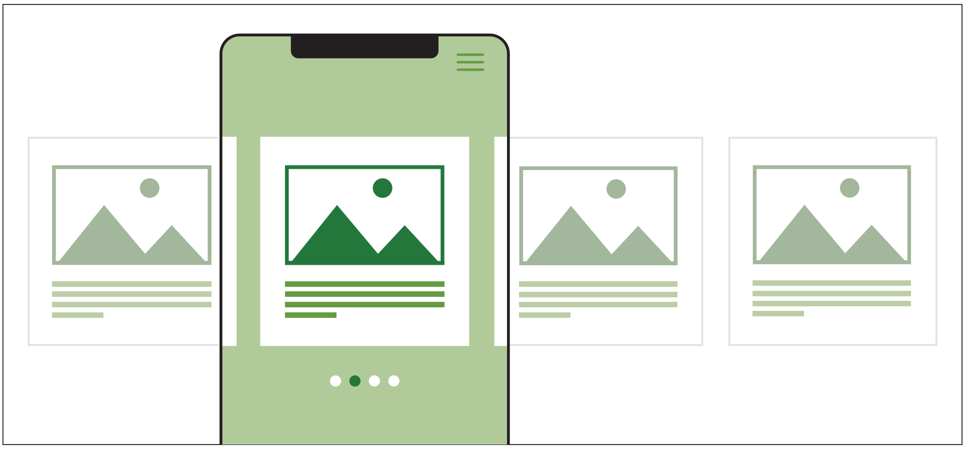


Figure 11.71 Modern carousel where only one content item is fully visible to the user at a time.

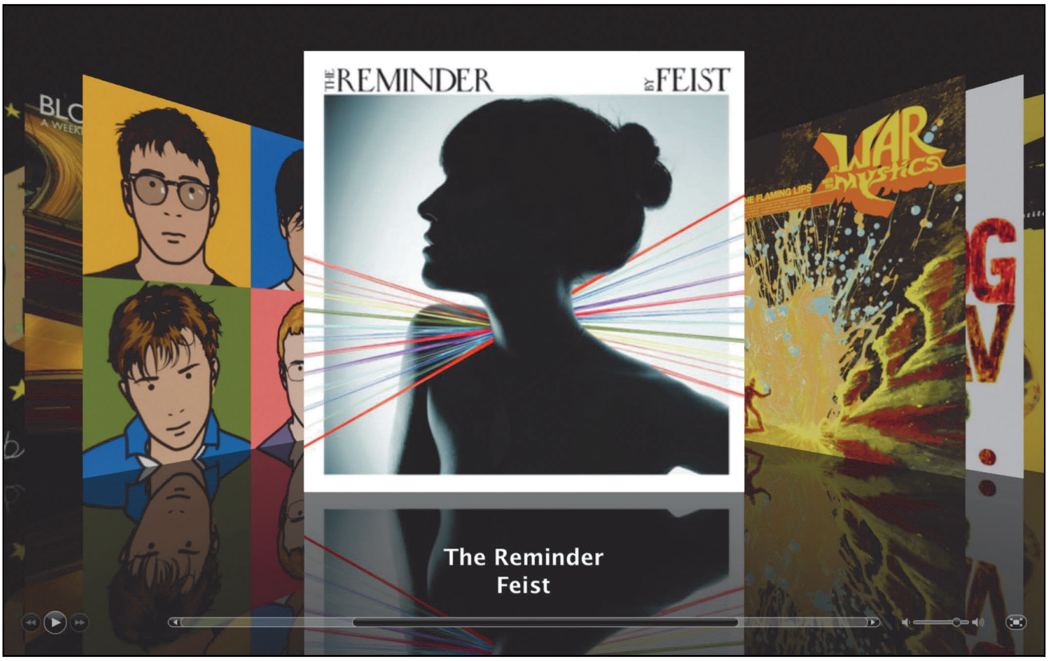


Figure 11.72 Cover Flow as used by Apple (https://en-academic.com).

Users can navigate through a carousel in various ways. One touch display variant moves the elements with the finger, like turning a card rack at the grocery store. Another variation is to interact by sliding a scroll bar, which then moves the images forward or backward. A final common option is to display arrows on the right and left edges, each of which can be clicked to move one image forward or back, as shown in Figure 11.73.

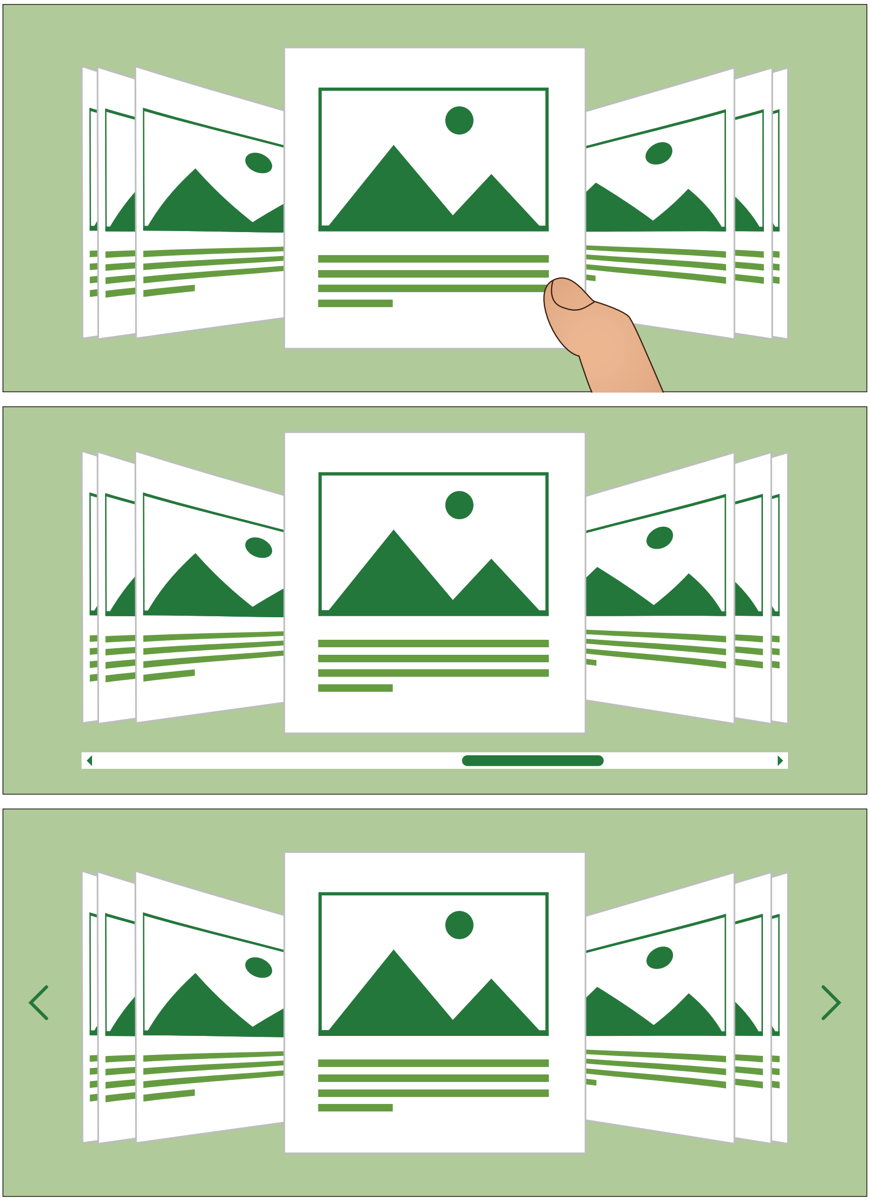


Figure 11.73 Various interaction options with carousels.

Compared to displaying different images side by side or one below the other, a carousel requires relatively little space since the images not currently focused are usually displayed diagonally to the sides, entirely as a line or not at all. A carousel should only be used when the different images do not need to be viewed in parallel or when it does not matter which image the user is viewing. A carousel is thus more of an element used playfully and interactively and less as a fixed workflow component.

In some apps, video content is also presented in carousels. In this area, a variant that we call the “infinite carousel” is becoming more and more popular. A popular representative of this series is YouTube Shorts where users can “swipe” vertically to see the next video. The unique feature in this case is that there is no end to the carousel.

11.6.17 Webpage Sliders

A webpage slider (we named this element in this way to distinguish it from the slider presented earlier in Section 11.6.9) is an element that displays information that changes according a specific rhythm and that can be actively controlled by the user, as shown in Figure 11.74. Sliders are often used as starting elements on web pages since they usually attract attention due to their large surface area and movement.

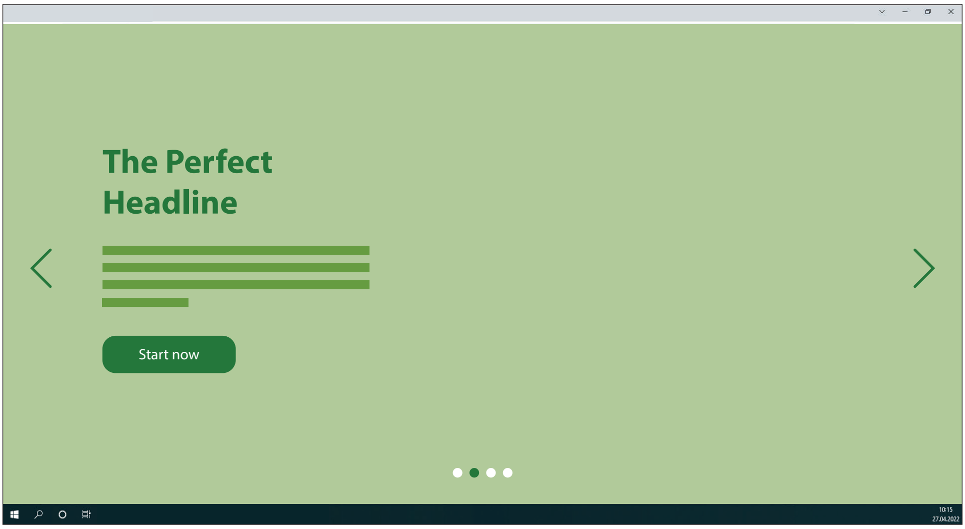


Figure 11.74 Typical slider element on the home page of a website.

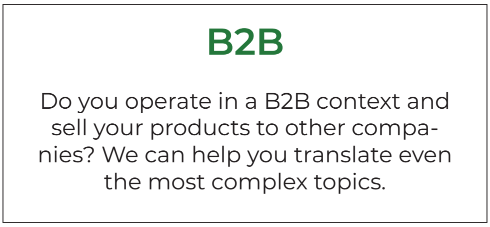
This slider element usually has small dots at the bottom, by which users can see which page the slider is currently on and how many pages are left in the slider. In addition to the display, sliders have arrows placed on the right and left edges of the screen. The arrows enable users to scroll through the information at their own pace or to retrieve a display that had been displayed earlier.

Ideally, the images or texts must be appealing and attract attention for sliders to work. However, here often lies one of the difficulties of slider elements. Since the content of the slider can be changed relatively easily without redesigning the entire page, we often observe that a wide variety of information is placed in the slider. This endeavor is not wrong and allows, for example, to share news or promote new offers very quickly. However, when implementing this kind of slider, make sure that the appealing look is maintained by selecting the right images, even when the content changes.

Additionally, only content relevant to users should be included in the slider. Since the slider element is usually quite large, it draws a lot of attention to itself. Therefore, all the content presented competes with the actual goal of the users on the page. Thus, we recommend using sliders in such a way that brings users closer to their goals.

11.6.18 Labels and Text Blocks

Label and text blocks are frequently used in designing applications and web pages. They do exactly what the names *label* and *text block* imply: They contain text and label other fields or are used for text content, as shown in Figure 11.75.



B2B

Do you operate in a B2B context and sell your products to other companies? We can help you translate even the most complex topics.

Figure 11.75 A typical text block.

The term *label* is usually used more for the labeling of input fields. *Text blocks* are typically multi-line output elements. Both elements are non-interactive but often contain essential information for the user, which is necessary for understanding and using the application. They can occur in different sizes: from short descriptive texts to detailed text elements.

Both elements are strongly determined in terms of design by the font, alignment, font size, and headings. For their display in a wireframe, however, a simple display of the text block or the label is usually sufficient or—if the content also needs to be tested—the pure text that should be included.

11.6.19 Images, Icons, and Placeholders

Another common interface element are images. So far, we've discussed images only as part of existing elements. Of course, images can also appear as their own individual elements. In a wireframe, images are usually represented as a free area that is crossed out starting from the corners—that is, with a *placeholder*, as shown in Figure 11.76.

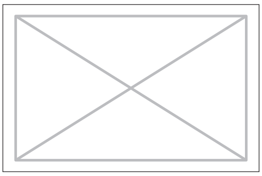


Figure 11.76 A typical placeholder in a wireframe.

This symbol is used to indicate the exact place where an image is to be seen later without having to define the content of the image already. The placeholder can also be used for other elements or areas that have not yet been defined. The display is then also shown, as shown in Figure 11.76. A specific placeholder representation is also used exclusively for images, as shown in Figure 11.77.

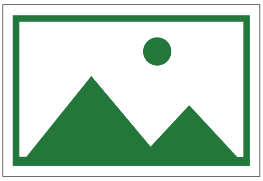


Figure 11.77 A placeholder especially for images.

Images play an essential role on websites. However, in many B2B applications, images tend to be used sparingly or only as part of the presented content. Since images can attract a lot of attention, before using an image, you should always consider whether an image will help users understand or will distract them from the task and content.

A big challenge is to select images that fit the *display format*. Especially in web-based applications with *responsive design*, often the visible section is not the desired image section since the image is automatically resized, as shown in Figure 11.78. In the case of *images with integrated fonts*, the text can quickly become illegible after a size reduction or can be cut off in the process.

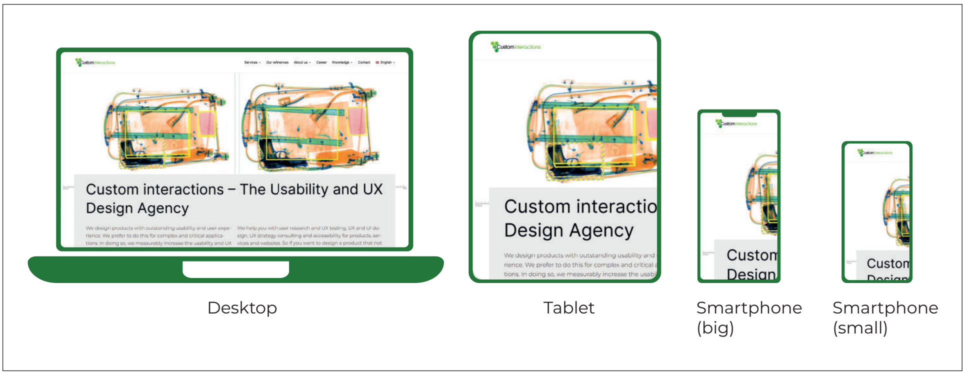


Figure 11.78 Poor responsive design where images with important content are cut off.

To use images to help users find their way around, explanatory icons are an excellent alternative to large images, as shown in Figure 11.79. Icons usually represent abstracted graphical information that is significantly less detailed but often provides visual orientation.

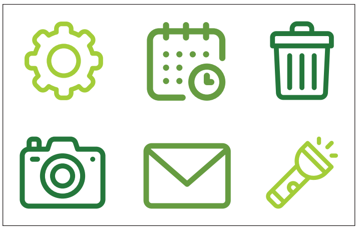


Figure 11.79 Various icons.

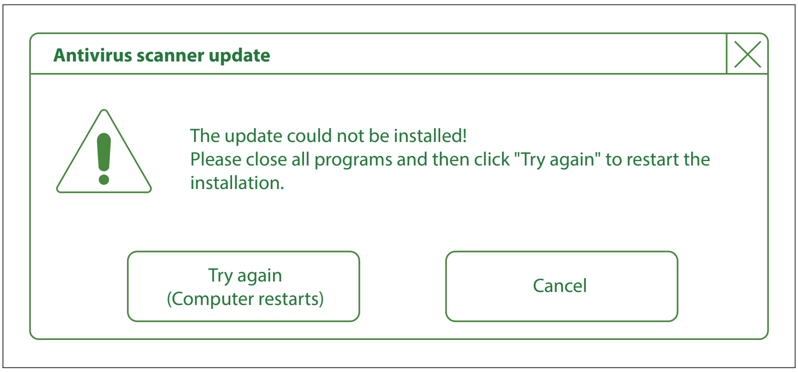
11.6.20 Special Case: Warnings

You need a *warning* whenever you want to inform the users about something that can have negative or indirect consequences. Warnings are unique because their content comprises the interface elements already presented. Still, in the best case, they follow a concrete structure to ensure that the users deal with the warning in the best possible way.

A kind of *syntax* should be followed when designing. Syntax is too fine a detail to be presented at the wireframe level, but since warnings are a kind of structural issue, we decided to include a discussion of syntax here anyway. Our recommendations are, on the one hand, based on *Atlassian Guidelines* (Atlassian Design Systems, 2023) and, on the other hand, based on the ISO 9241-13:1998 standard.

When designing a warning, as shown in Figure 11.80, you should consider the following elements:

- A meaningful headline
- A symbol matching the severity
- A description of the problem
- Recommendation actions to solve the problem
- Clearly labeled buttons



Antivirus scanner update

The update could not be installed!
Please close all programs and then click "Try again" to restart the installation.

Figure 11.80 Example warning that follows the structure described.

11.6.21 Conclusion

In this section, we introduced you to the most common interface elements based on ISO 9241-161:2016. With these elements, you should be able to design most UIs completely. Refer to the standard for the complete list.

At this point, we would like to conclude our section on wireframes and turn now to the question of how you can check the quality of the concepts we've just covered.

11.7 User Interface Quality Check: The Design Grid

A big challenge in digital applications is presenting a large amount of content in an appealing and clear way.

We often notice that beginners first start somewhere in the design and then add information or elements "where it fits" piece by piece. Unfortunately, this approach often makes the overall picture of the screen look chaotic later and prevents users from easily following along with and moving through the content effortlessly. However, what might be entertaining in a hidden object game is often off-putting in digital applications: A large amount of information, no clear clues as to what information can be found, no structure to shimmy along? Well, that all leads to frustration instead of a good UX.

At this point, we would like to introduce working with a *design grid* as a simple solution. The grid divides the screen into different areas and helps you align elements uniformly. The grid corresponds strongly to the human need for clear boundaries and shapes.

In their book, Heimann and Schütz (2018) describe that people tend to restrict their territories, often in rectangular shapes. This behavior can be observed with a towel on the beach, carpets, or a fence framing a property. Suppose you ask about the associations attached to rectangular shapes. In this case, these associations are often described as "static" and "constant." In addition, clear areas help to group information, making it easier to grasp (recall the *Gestalt principles* described in Chapter 7, Section 7.4.3). Already, many elements in the digital world are often rectangular. Images or text blocks are exactly such elements.

The design grid plays perfectly with implementing web content, as this concept is ideally suited for responsive web design (Section 11.11) and thinking about structures and blocks. But what exactly is the design grid?

The simple answer: The design grid is simply vertical and horizontal guides you place in your application during design. You use these guidelines to define page margins, menu areas, and the beginning or end of an interface element. In this way, you can ensure the screen maintains a specific basic structure during design by aligning all elements with the guides. You can also use the design grid to evaluate the quality of a design easily and quickly.

First, let’s look at using the *vertical guides*. Take your design and track through the inserted elements from left to right, as shown in Figure 11.81.

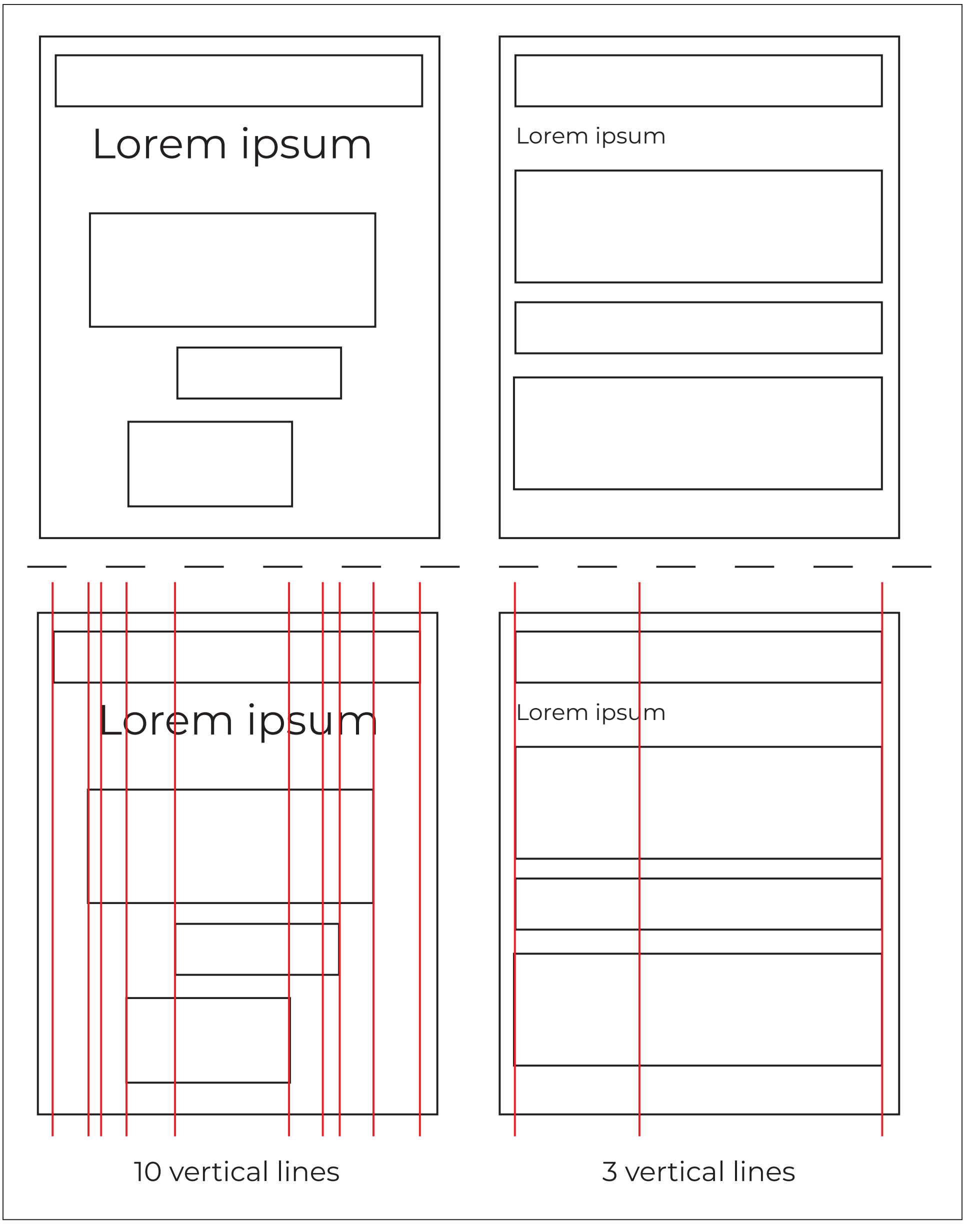

Lorem ipsum
Lorem ipsum
Lorem ipsum
Lorem ipsum
10 vertical lines
3 vertical lines

Figure 11.81 Our design grid in action (part 1): A choppy wireframe exists in the upper left, and a much calmer-looking wireframe on the top right. Vertical grid lines have been placed on their elements.

Draw a vertical line on the left and right edges of each element. In our example, the result is an entirely different image, even though the elements shown are the same in content. The goal of the design grid is to minimize the vertical lines because the more vertical lines you have, the choppier your interface looks.

The horizontal lines in our design grid follow a similar rule initially. You go from top to bottom through the inserted elements and place a horizontal line at the top and bottom edges of each element, as shown in Figure 11.82.

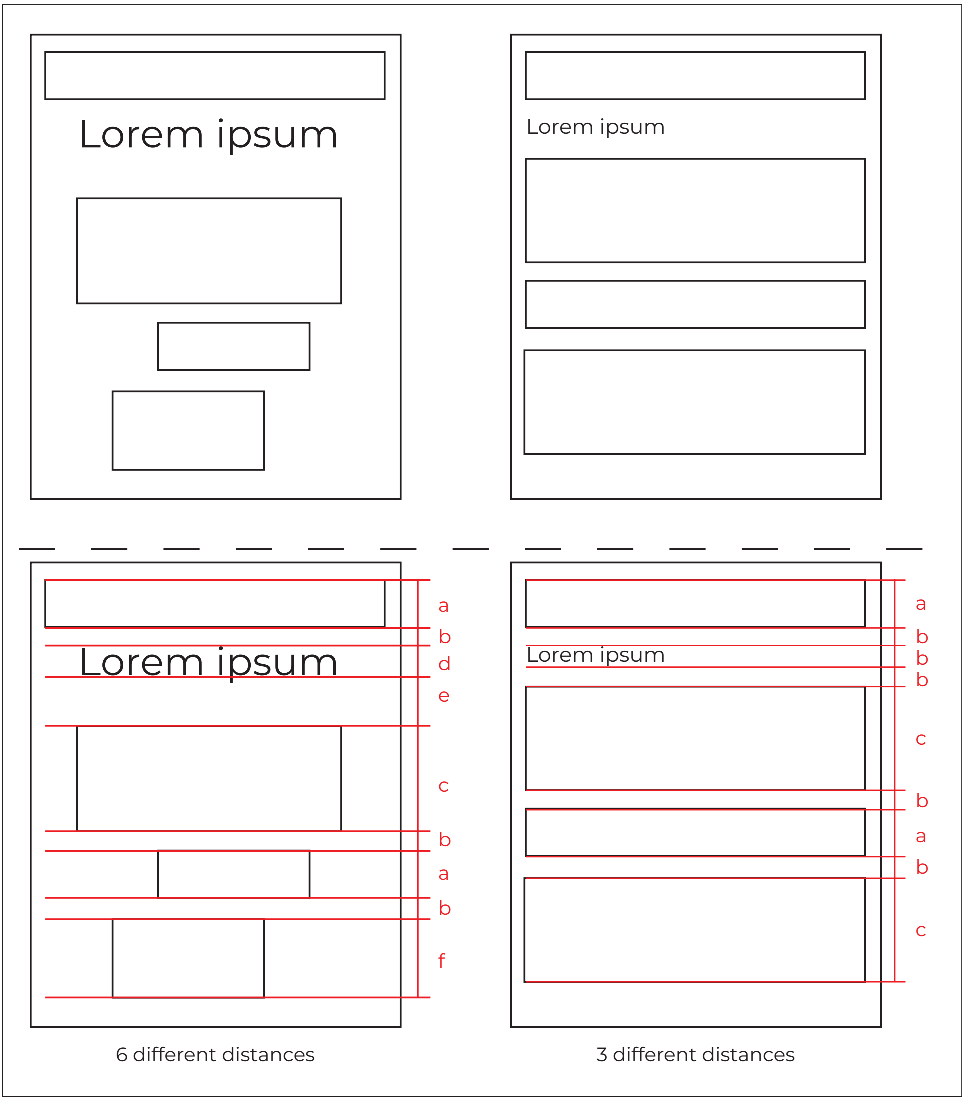

Lorem ipsum
Lorem ipsum
Lorem ipsum
a
b
d
e
c
b
a
b
f
Lorem ipsum
a
b
b
b
c
b
a
b
c
6 different distances
3 different distances

Figure 11.82 Our design grid in action (part 2): Our choppy wireframe compared to the calmer wireframe. Horizontal grid lines are added, and the distances between them are determined.

Once you have created these lines, you can judge (or measure, if you want to be precise) the distances between the lines. To create a smooth image in your design in this direction, minimizing the number of different distances is now the task at hand.

You can use the design grid not only to advance your design, but also to efficiently perform a before-and-after comparison to show whether and how much you were able to optimize the structure of the screen. Of course, since the design grid is based on the elements in the interface, you can (and should) use this concept as early as wireframing. However, evaluating and checking alignment also works with finished screens, as shown in Figure 11.83.

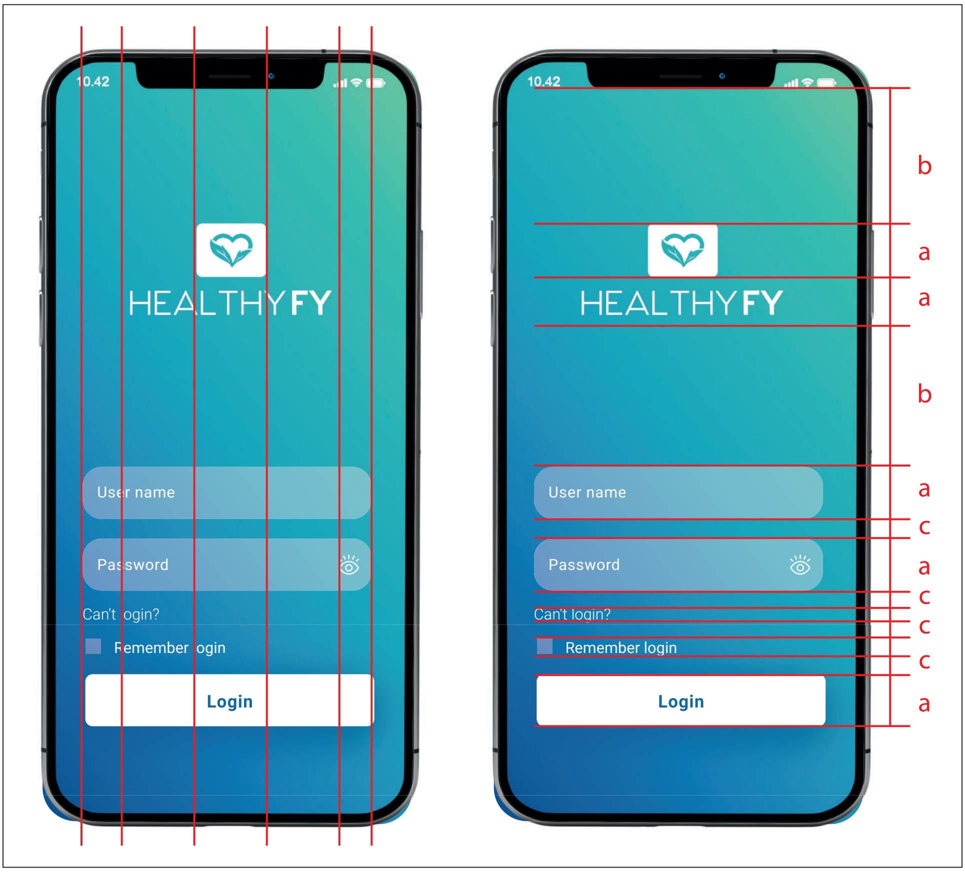

10.42
HEALTHYFY
User name
Password
Can't login?
Remember login
Login
10.42
HEALTHYFY
User name
Password
Can't login?
Remember login
Login
b
a
a
b
a
c
a
c
c
c
a

Figure 11.83 Our design grid in action (part 3): deployment on a visually elaborated UI.

Working with the grid sounds boring—we are aware of that. However, you bring order to the information through the grid and prepare your content to fit perfectly with the responsive design, which also works with grids.

But suppose you want more movement and more dynamics in your designs. In this case, no problem: You can get movement by working with a background image and animations and by designing individual interface elements. Of course, you can also deliberately "break out" of the grid at defined points to set accents. A best practice is to define the grid as early as possible and always refer to it during evaluation. You'll notice: Once you've practiced thinking this way, you'll be able to spot deviations from just a few pixels away from a relatively long distance. Promise!

11.8 Screen Flows

Everything we've presented so far involves creating and evaluating individual wireframes. We've discussed interface elements and the grid to check if the alignment fits. So far, however, we haven't talked about *interaction*. Unlike print, in usability and UX design, you're not designing a document—you're designing an interaction.

This premise automatically assumes you must consider the individual wireframes, their content, and how a user will move through your application. This movement needs to be planned. Before you start designing the individual screens, we encourage you to plan out the workflow (see, for example, Chapter 9, Section 9.4.1, and Chapter 10, Section 10.3). Now, we'll come back to this planning.

Initially, the workflow may have been rough and consisted of simple boxes. Now, you fill in these boxes with life—or rather, with wireframes! The results of this activity are called *screen flows*.

When discussing screen flows, we mean showing the interaction paths based on the created wireframes. Screen flows are utilized to illustrate how users move from one screen to the next. For this purpose, arrows are attached to individual interaction elements to show clicks and where the user ends up, as shown in Figure 11.84.

Sometimes, screen flows are also called *user flows*. No matter what you call them, at this stage, the important point is that not only are theoretical steps documented, as in the case of workflows at the beginning, but the steps are based on the actual wireframes. In this task, we are tracing a user's movement on and through our screens. Understanding this "movement logic" is vital when developing a real application from the wireframes later on to avoid guessing how individual screens merge into one another. Screen flows can thus serve as a blueprint for developers and are often a precursor to *click dummies*, which we'll discuss in the next section.

In addition to representing a screen flow through arrows, other display variants are also possible, for example, numbering the individual elements and assigning new screens to the numbers, as shown in Figure 11.85.

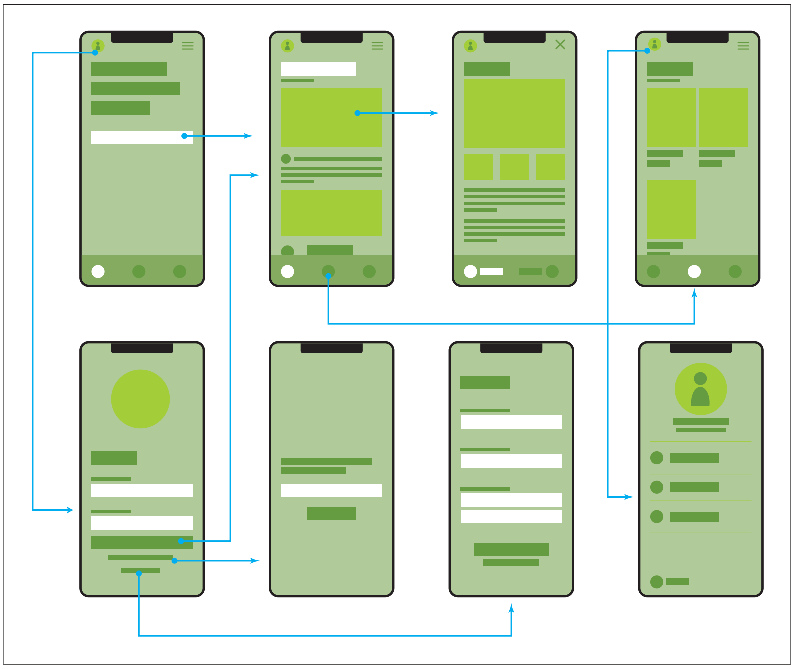


Figure 11.84 Example of a screen flow with wireframes.

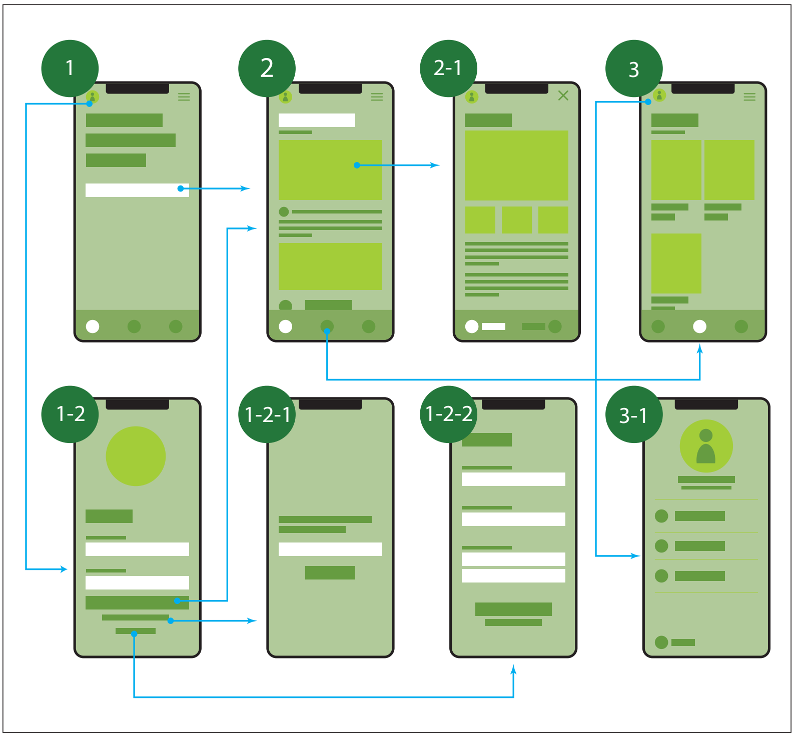


Figure 11.85 Alternative representation of screen flows.

11.9 Click Dummies

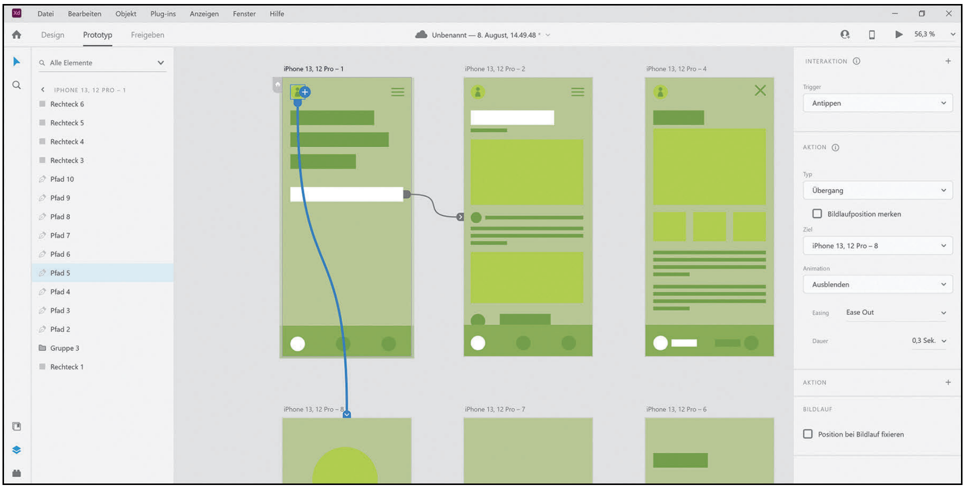
So far, we have created individual screens and theoretically "lined them up" by specifying their order and specifying their movement logic. When you present a screen flow for evaluation by a group of users, most people will have a tough time imagining how using the tool will feel to them in the end. That's a perfectly normal response and also why click dummies are so important for evaluating workflows and their usability and UX.

Click dummies are the interactive form of screen flows. Instead of showing multiple wireframes in an overview, a click dummy behaves like the actual application. Thus, the individual controls are usable, and you can move through the application. In other words, a click dummy is an *interactive prototype*.

This reference helps users imagine how the interaction with the application feels. Determining whether all information is available and all required functions have been integrated becomes straightforward. You can also check whether users can find and understand important information independently.

When creating click dummies, you can simply follow the construction manual of your screen flow. Depending on the tool, the implementation is complicated in different ways. In simple tools, you often place an invisible click area over individual interface elements and define to which wireframe the click dummy jumps to after clicking the button. These jump areas are not visible to users, only the wireframe interface elements. This capability allows you, for example, to insert "secret" buttons that are only relevant for the execution of tests (e.g., triggering error states).

Some tools link the interaction directly to the control element, so integrating additional surfaces is no longer necessary. For example, you can simply drag and drop the interaction paths to the next wireframes, as with the blue line shown in Figure 11.86.


Datei
Bearbeiten
Objekt
Plug-ins
Anzeigen
Fenster
Hilfe
Design
Prototyp
Freigeben
Unbenannt — 8. August, 14.49.48 *
56,3 %
Alle Elemente
IPHONE 13, 12 PRO – 1
Rechteck 6
Rechteck 5
Rechteck 4
Rechteck 3
Pfad 10
Pfad 9
Pfad 8
Pfad 7
Pfad 6
Pfad 5
Pfad 4
Pfad 3
Pfad 2
Gruppe 3
Rechteck 1
iPhone 13, 12 Pro – 1
iPhone 13, 12 Pro – 2
iPhone 13, 12 Pro – 4
iPhone 13, 12 Pro – 8
iPhone 13, 12 Pro – 7
iPhone 13, 12 Pro – 6
INTERAKTION
AKTION
AKTION
BILDLAUF

Trigger: Antippen

Typ: Übergang

☐ Bildlaufposition merken

Ziel: iPhone 13, 12 Pro – 8

Animation: Ausblenden

Easing: Ease Out

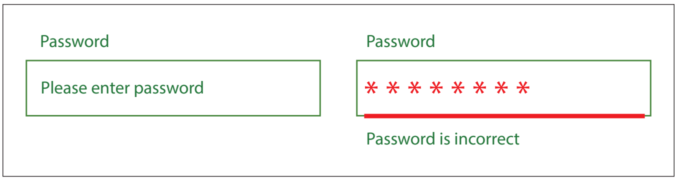
Dauer: 0,3 Sek.

☐ Position bei Bildlauf fixieren

Figure 11.86 Click dummy in Adobe XD.

Creating the wireframes with the same tool as the click dummies saves you a lot of work because you can continue using your existing wireframes. Not only is it possible to switch between wireframes with a click, but button states can also be changed with a click, as shown in Figure 11.87.

How detailed you make your wireframes and, thus, also how detailed your click dummies are depends heavily on your goal. The closer you are to implementation, the better you can predict what will work and what won't. At a certain point, however, the effort for prototyping exceeds the actual creation effort.



Password is incorrect

Figure 11.87 Elements can have different states in most click dummy tools.

11.9.1 Tools for Creating Click Dummies

Immense importance lies in choosing the right tool for creating click dummies to make their creation as fast and easy as possible. In this section, we'll provide a brief overview of some tools we've successfully used to design click dummies. Of course, you can use many other tools, but the following three have proven to be highly flexible and powerful:

- Axure RP
- Adobe XD
- Figma

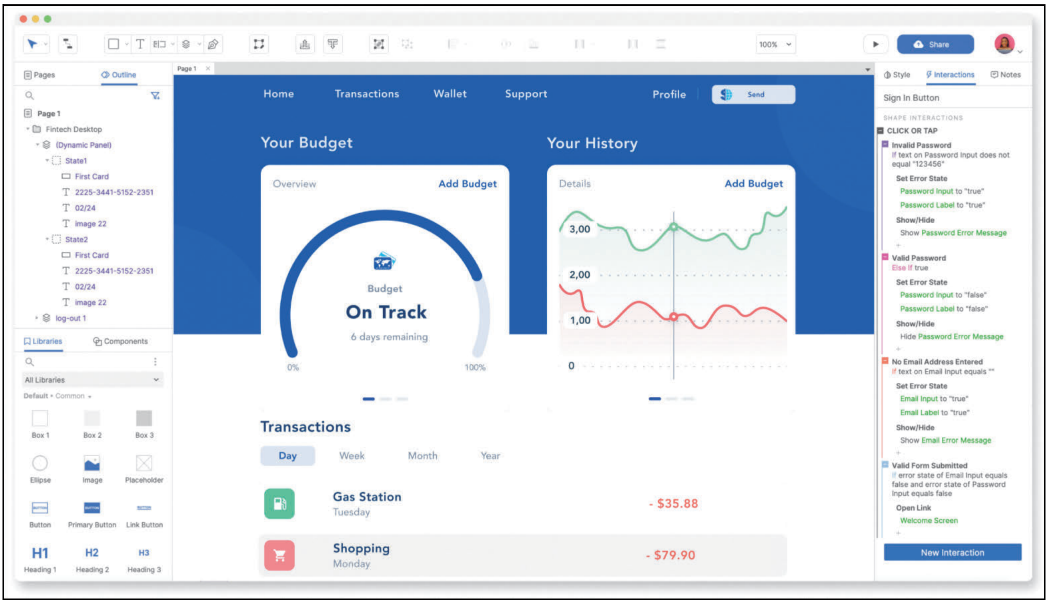
In this section, we'll focus on each tool's advantages and disadvantages. We won't present all the shared aspects of each tool: For example, all tools allow a click dummy to be shared via links, thus allowing a click dummy to be used on other devices in remote testing. Another common feature is the ability to store element libraries and thus rebuild your own (or someone else's) existing design system.

Please be aware that the functionality of software tools changes relatively fast. Therefore, evaluate these tools with the most up-to-date information before deciding on the ideal one for your needs.

But now, let's get into a short overview.

Axure RP

Axure RP, shown in Figure 11.88, is one of the most powerful prototyping tools. From creating individual wireframes to completing interactive click dummies that look like finished products, anything is possible with Axure RP. In addition to enabling the visual design of click dummies and their interactivity, individual elements can even be implemented to use simple programming logic. Thus, you can create detailed interactions based on a user's action and enable the output of errors—not at the push of a button—but in reaction to a user's "incorrect" action.


Pages
Outline
Page 1
Fintech Desktop
(Dynamic Panel)
State1
First Card
2225-3441-5152-2351
02/24
image 22
State2
First Card
2225-3441-5152-2351
02/24
image 22
log-out 1
Libraries
Components
All Libraries
Default • Common
Box 1
Box 2
Box 3
Ellipse
Image
Placeholder
Button
Primary Button
Link Button
H1
Heading 1
H2
Heading 2
H3
Heading 3
100%
Share
Home
Transactions
Wallet
Support
Profile
Send
Your Budget
Overview
Add Budget
Budget
On Track
6 days remaining
0%
100%
Your History
Details
Add Budget
3,00
2,00
1,00
0
Transactions
Day
Week
Month
Year
Gas Station
Tuesday
- $35.88
Shopping
Monday
- $79.90
Style
Interactions
Notes
New Interaction

Sign In Button

SHAPE INTERACTIONS

CLICK OR TAP

Invalid Password
If text on Password Input does not equal "123456"
Set Error State
Password Input to "true"
Password Label to "true"
Show/Hide
Show Password Error Message

Valid Password
Else If true
Set Error State
Password Input to "false"
Password Label to "false"
Show/Hide
Hide Password Error Message

No Email Address Entered
If text on Email Input equals ""
Set Error State
Email Input to "true"
Email Label to "true"
Show/Hide
Show Email Error Message

Valid Form Submitted
If error state of Email Input equals false and error state of Password Input equals false
Open Link
Welcome Screen

Figure 11.88 Axure RP.

You can display tables with filter options or create adaptive views for different devices, meaning you don't need to create multiple click dummies for different devices: A click dummy automatically adapts to the device.

Another advantage? Axure RP can import files from Adobe XD and Figma. We always use Axure RP when we "need to go that extra mile." Some examples include if we want to show some particularly sophisticated interactions, if the user needs to feel that data is being derived from somewhere specific, or if errors need to be mapped realistically to understand the consequences of actions. For example, you could create logons where the user enters data or search boxes where only specific search terms actually "work."

Axure RP is mighty and, therefore, more complex than the other tools on our list. The Axure team has taken many opportunities to respond to market demand over the last 20 years. As a result, Axure RP is a bit more technical than the other two prototyping software solutions we discuss in this chapter. Nevertheless, Axure RP does not require programming expertise until the point at which the implemented logic itself becomes complex. Most UIs in Axure will require no programming at all.

Formally speaking, Axure RP could be considered the "top dog" of UX design tools, and its focus lies on interaction. In contrast, the other tools operate more in the UI design space, where the focus lies on interface design.

Adobe XD

Adobe XD, as shown in Figure 11.89, is a prototyping tool from the Adobe family, meaning the tool is ideally linked to other suite applications like Adobe Photoshop or Adobe Illustrator. This integration is an advantage over the other two tools, especially when working extremely visually. Even though we are still in the realm of click dummies with Adobe XD, we consider great visual design as vitally important since additional effort can be avoided later if the rendering is done in the same tool as the wireframing.

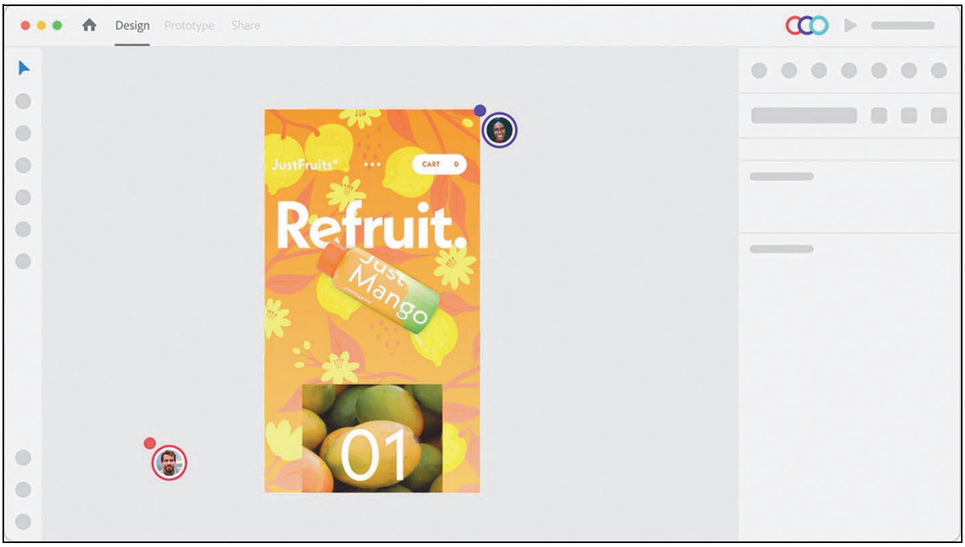


Figure 11.89 Adobe XD.

Adobe XD has an evident design focus. The goal is to create visually superior click dummies that exploit the full potential of the design. Nevertheless, implementing interactions and switching states is also possible in Adobe XD. Adobe XD even supports the implementation of interfaces using voice control.

Overall, Adobe XD is a bit faster and easier to use than Axure RP and even more accurate the more you already use the Adobe suite. On the other hand, the scope for designing realistic interactions is significantly limited. Ultimately, Adobe XD has been catching up significantly in this area in recent years, offering increasingly more features for interaction design. However, its capabilities cannot yet match Axure RP, especially for dynamic content.

Figma

Figma, shown in Figure 11.90, is a relatively new tool that has significantly benefited from the work-from-home trend—especially recently. Figma focused on collaborative work in teams at an early stage and implemented this capability much better than the tools before it.

With Figma, collaboration becomes possible because changes are saved live in the cloud. Even during editing, several people can create click dummies and prototypes simultaneously. A marker on the screen also shows all the participants who are currently working and where they are working. This capability was recently added to Adobe XD—but Figma is still more sophisticated in collaboration.

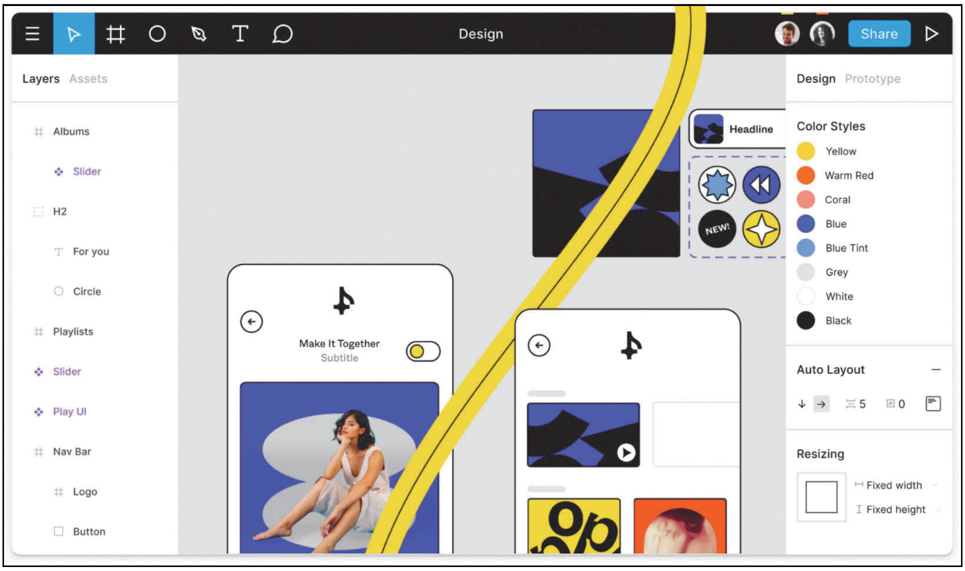

Design
Share
Layers
Assets
Albums
Slider
H2
For you
Circle
Playlists
Slider
Play UI
Nav Bar
Logo
Button
Headline
NEW!
Make It Together
Subtitle
Design
Prototype
Color Styles
Yellow
Warm Red
Coral
Blue
Blue Tint
Grey
White
Black
Auto Layout
5
0
Resizing
Fixed width
Fixed height

Figure 11.90 Figma.

The advantage of Figma is, unfortunately, also a disadvantage. Ideally, the files are stored in the cloud for this collaborative work to function. Therefore, we only use Figma when a customer expressly requests its use and agrees to meet all data protection and confidentiality requirements. We recommend you find out in advance what requirements your IT department or your customers have regarding your tools.

Figma offers an easy introduction to click dummy creation and helps you find the best possible workflow in the early stages through simple workflows within the tool. Like Adobe XD, Figma emphasizes ease of access and offers simple interactions that can be quickly integrated into prototypes. However, the greater the complexity of the click dummies gets, the more likely you'll find yourself back in Axure RP.

11.9.2 Making the Design Testable

Whatever tool you choose, the goal is always to make the design testable. By "testable," we mean that you build the click dummy so that all the cases you want to evaluate later in a usability and UX test are testable through the click dummy.

For this purpose, you must consider what you'll test early on. Often, click dummies are designed in advance so that the interaction is understandable but still does not represent the actual workflow, only a simplified one. This simplification makes conducting a user test difficult, especially when considering the user's point of view regarding data appearing automatically in the test or the tool advancing in steps that the user (under normal circumstances) would take over.

As a general rule, you should remember: Even if the click dummy doesn't look finished, the interaction with the user should be as finished as possible. Only in this way will you detect and eliminate potential stumbling blocks early on.

Of course, this advice only applies to the area you want to test with users. You can omit or simplify all areas "outside" that range to your heart's desire or insert placeholders to lead users back to the desired test area.

11.10 Desktop, Mobile, or Smartwatch First?

At this point, we hope we've provided a good overview of the steps you need to take to define an interaction structure for an application. However, we're not at the end of our story and want to draw your attention to a topic that has become increasingly important in recent years: designing applications for mobile devices.

"*Mobile first*" is the increasingly widespread design motto in use. In this section, we'll first look at this approach and show why, in some areas, "mobile first" makes sense. However, at the end of this section, we'll highlight some other approaches that are helpful in other situations. But let's start with "mobile first."

Recently, mobile display devices such as tablets and smartphones have become increasingly important. The focus has shifted significantly toward mobile display devices in the web context. In 2011, approximately 6% of global Internet traffic was mobile, but in 2017, for the first time, mobile Internet traffic narrowly surpassed desktop traffic, accounting for more than 51% of accesses, as shown in Figure 11.91.

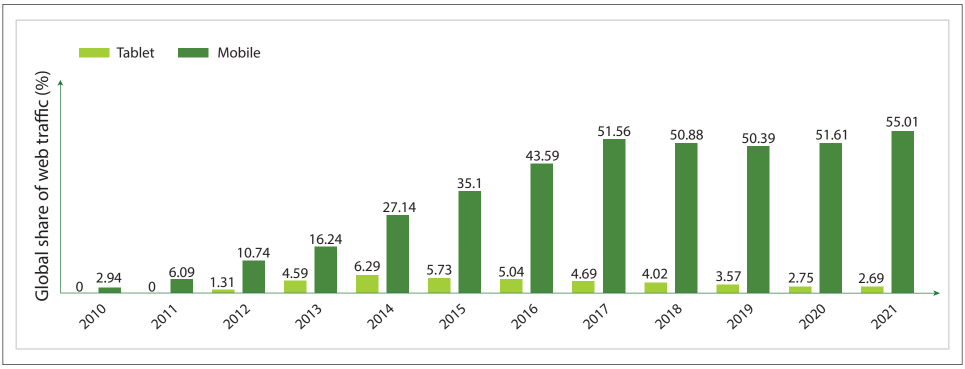


Figure 11.91 Global share of web traffic via cell phones (based on data from https://gs.statcounter.com/).

In 2021, more than 55% of web traffic came from *smartphones*, and we expect this trend to continue. "Mobile first" is, therefore, more relevant today than ever. And this belief doesn't just apply to the private sphere. Work applications were once almost exclusively on desktops, but the number of web clients—and thus the proportion of possible mobile use—has increased significantly in recent years.

In addition to this shift in the use of already well-established forms of access, *smartwatches* have come increasingly into focus in recent years. In the US alone, smartwatch users increased by nearly 50% between 2017 and 2021, from 21.1 million to 41.6 million (*www.statista.com*). There is still no end in sight in this market, which rapidly creates new possibilities in the design of applications because suddenly a large amount of data is available that was previously not accessible or only accessible at great expense. On the other hand, the available display is again reduced enormously.

Typical Screen Sizes

The *Statcounter* website (*https://gs.statcounter.com/screen-resolution-stats*) provides the following information (as of 2023) about the most common screen resolutions:

- Desktop displays range from 1024 × 768 to 1920 × 1080 pixels.
- Mobile displays range from 360 × 640 to 414 × 896 pixels.
- Tablet displays range from 601 × 962 to 1280 × 800 pixels.

In addition, around half a million users were analyzed to identify the most frequent resolutions for desktops or laptops in 2021. The four most commonplace resolutions, presented in descending order of frequency, are the following:

- 1920 × 1080 pixels (19.85%)
- 1366 × 768 pixels (14.31%)
- 1536 × 864 pixels (9.45%)
- 1440 × 900 pixels (6.23%)

Significant differences can already be seen with these figures. If smartwatches are integrated, as in the case of the *Apple Watch*, the resolutions differ even more. The smallest versions of the display for the Apple Watch include the following:

- Apple Watch 1 to 3: 272 × 340 pixels
- Apple Watch 4 to 6: 324 × 394 pixels
- Apple Watch 7: 352 × 430 pixels

The range of resolutions is now quite wide, with no end in sight. Wearables increase at the resolution scale's lower end is not enough. On the other hand, increasingly smart TVs are emerging that allow Internet access and the display of web content. Newer TVs and PC monitors have 4K resolution, corresponding to 3840 × 2160 pixels, or even 4096 × 2304 pixels with 4K2K resolution. In addition, the next evolutionary stage is at the starting gate with 8K TVs, which can offer resolutions up to 7680 × 4320 pixels.

Another complicating factor is that, for some applications, you must use several devices or screens in parallel, for example, in the fitness sector.

With *Apple Fitness+*, for example, a user often discovers and selects a workout by browsing the smartphone app. The actual workout, however, runs on the TV or tablet because the larger screen is better for watching the trainer demonstrate exercises. In addition, live fitness data is also displayed on the TV (i.e., the heart rate). This data comes live from the Apple Watch on your wrist. Thus, a wearable functions both as a data provider and as a kind of remote control. The device can be used to start, stop, and pause workouts. The UX is not limited to one device or display but extends over several devices in a single use case, as shown in Figure 11.92.

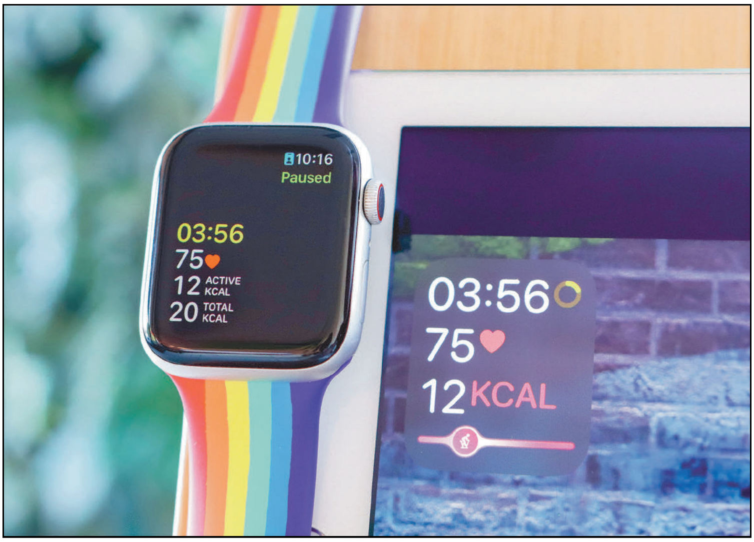


Figure 11.92 Apple Fitness+ uses multiple devices for one use case: fitness training.

As a result, a wide variety of resolutions must be considered in usability and UX design, and each resolution exclusively wants to be operated in meaningful way. This approach is called *responsive design*, which—due to its importance—we'll dedicate a separate section to (Section 11.11).

But to return to the question of this section, "Which one first? Desktop, mobile, or smartwatch?" Our answer is as straightforward as it is simple: "It depends."

Design is the third of four steps in our data-driven UX design process. Although you'll most likely go through each step multiple times, at this point, you should already have a good idea of who you're designing the application for and what the environment, tasks, and user behavior might look like. On this basis, you'll make your decision by asking the following questions:

- Are you designing an internal application that is used exclusively on the PC? Then, "mobile first" makes little sense.

- Is it a web application? Then, you should definitely think “mobile first.”
- Are you perhaps even in the fitness sector? Then, “wearable first” or “smart TV first” applies.

And thus the series continues: You might have an *embedded system* with a fixed monitor setup. Optimize for that embedded system. We hope this discussion makes the principle clearer.

But what exactly does that mean? The idea behind the “-first” approach is to design primarily for the target format on which the content should be accessed. In the case of “mobile first,” you should consider how the application will look on a smartphone before considering other formats, such as desktop applications. If two or more display formats are to be operated in parallel, start your design journey from the smallest format. Adding something to an application is always easier than taking something away.

Above all, you should avoid falling into the trap of wanting to display every piece of information in every format. The smaller your display, the more selective you must be when *choosing content*. But that’s not a bad thing. Rarely do users want to do precisely the same thing with a smartwatch as they do with a desktop PC. The key is to have the courage to reduce and, above all, to test with users. If you try to keep all applications the same, it just won’t work—we promise.

Thus, your first question should be, “What information and interactions do users need on the device?” Next, cross out anything that you feel is nice to have but not necessary. You can always expand later once you’ve presented the core information in a simple, clear, and easy-to-understand way.

As mentioned earlier, the presentation of the content must differ between different formats to ensure optimal usability and UX. To achieve this goal, you can now *design individual applications* for each medium and tailor these applications to meet a display’s exact needs. This coordination is a great way to ensure the best fit between content and display medium. However, designing a custom application also means putting multiple effort streams into creating and maintaining the applications. You also run a risk of changes and updates not being reflected in all versions because people forget to transfer these changes. There must be a better way!

11.11 Responsive Design

Responsive web design is an excellent way to better address the issue of different screen resolutions we described in the previous section, at least for web applications. Responsive web design is about the idea that the created layout is divided into different elements, which are then arranged differently depending on the display medium, omitted entirely, or changed to look completely different. In this way, the layout can flexibly react to the requirements of the display device or to the user requirements with this display device, thus ensuring better display quality. Instead of the same web page being squeezed into a small screen, a responsive design arranges the content differently.

The design grid, which we described in Section 11.7, is handy for the implementation of responsive design. With the grid, you can arrange interaction elements into fixed areas. The same arrangement happens with responsive design, except the pagination is also defined. Let's demonstrate this principle through an example.

If the screen shown in Figure 11.93, on the left, is your application in its desktop variant, then you can see how the images are arranged side by side, with their labels in the row below. The same application as a mobile version might look like the screen shown on the right. Notice how the images are below each other, with their texts, so that the size of the individual images can remain almost the same size or can increase in size relative to the screen.

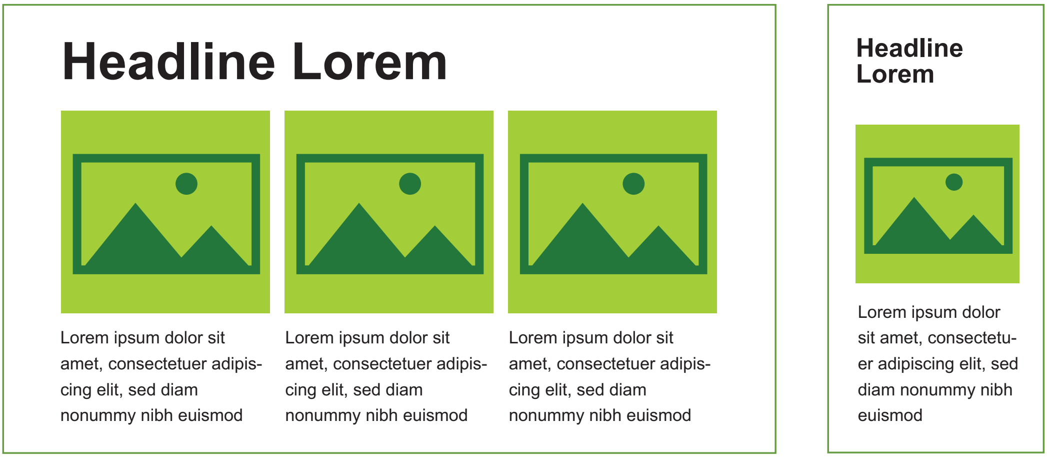

Headline Lorem
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
Headline Lorem
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod

Figure 11.93 Responsive design for a web application (left) and smartphone (right). Notice responsiveness is more than a size reduction, but instead the content is optimized for the screen size.

We won't delve more deeply into the details of responsive design at this point. Various books cover this topic in much more detail than we can in this book. We recommend especially *Responsive Web Design with HTML5 and CSS: Build future-proof responsive websites using the latest HTML5 and CSS techniques* by Ben Frain.

As always, however, you should test a design with users afterward. When selecting resolutions and devices, you should look at your target groups. Which resolutions and devices do your target groups (mainly) use? In this way, you can determine whether the elements work for the device class and whether the workflow fits the class. We present the methods for these tests in detail in Chapter 12.

11.12 Visual Design

Visual UI design and development occurs parallel to the development of interactions. Even though this book focuses primarily on usability and UX design, we want to expand our approach briefly to developing visual design since the visual design also has a significant influence on the subsequent UX.

We can recommend various books if you want to get to grips with visual design in detail. Particularly noteworthy are the books *Graphic Design School* by David Dabner et al. and *Graphic Design Theory* by Meredith Davis. In our presentation, we assume that you have mastered these basics or are still acquiring them. Thus, we'll focus only on the process and individual steps we go through to achieve good results with our customers.

11.12.1 Visual Design Research

In Section 11.2, we start with *research* before creating our first wireframes. Among other things, we look at how competitors design their solutions and position them on the market. This information influences our UI concept as well as our visual design.

If you've already conducted sound design research at this point (which should be the case), no further steps are probably necessary. If not, now is the time to look at your research results, pull out the findings (described in the remainder of this section), or add to them accordingly.

Visual design research is about getting a feel for the client's style, the industry's style, current trends, and applications.

The Style of the Customer

Most likely, the most important criterion for choosing a suitable visual design is the (desired) style of the customer. The customer can be your company or an external customer, if you're a service provider. *Style* can either mean an already-defined style or a desired future style. If you already have a perfect sense of visual style, then you can skip this part. Otherwise, start by "browsing" everything communicated externally by the client or what design documentation is available, including the following information:

- The website of the company
- The company's style guide or design system (if available)
- The design of existing products of the company
- The design of other corporate communications (e.g., flyers, social media posts, trade show booths, etc.)

Feel free to add other items to the list as you come across them. We only suggest some areas you should look at.

Then, check the various sources for similarities, color languages, form languages, and how the user is addressed. Does the company communicate formally or informally? Is content communicated actively or passively? On this basis, you can develop a feel for how the company wants to be perceived from the outside.

Of course, you can also talk to people in the company who are responsible for external communications on this point. Depending on the company's size, we typically talk to management or those responsible for marketing.

The Style of the Industry

The same thing we just did for our own company or a client company, you can do for the industry in general. In most industries, a uniform language of form and color exists so that all interested persons can quickly understand whether the company "belongs" to an industry or not. Consider these two examples to illustrate this principle:

- Regarding "sustainability," the color language usually includes green or green/brown with a more organic design language, as shown in Figure 11.94. The goal is about visually representing being "close" to nature.

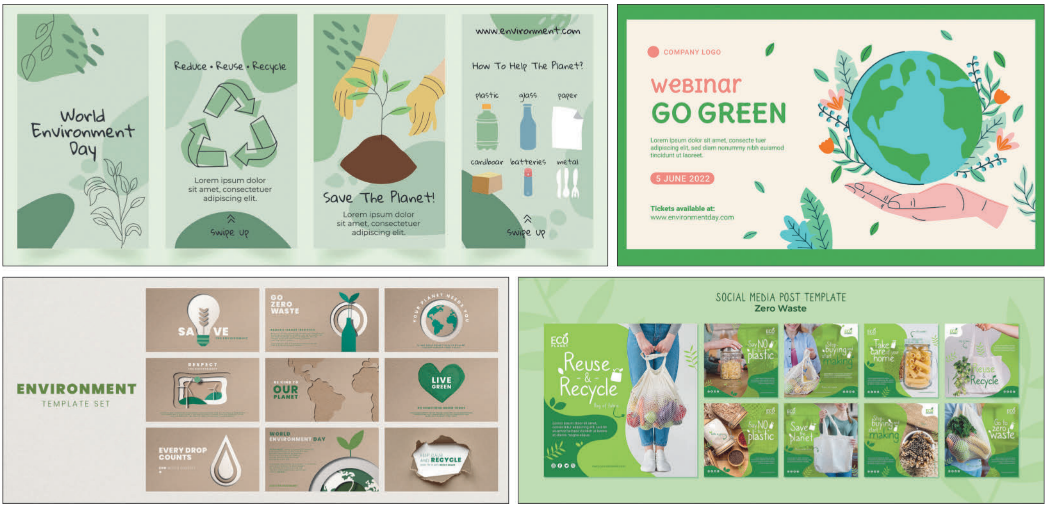

World Environment Day
Reduce • Reuse • Recycle
Save The Planet!
www.environment.com
How To Help The Planet?
Swipe Up
COMPANY LOGO
webinar
GO GREEN
5 JUNE 2022
Tickets available at:
www.environmentday.com
ENVIRONMENT
TEMPLATE SET
SAVE
GO ZERO WASTE
LIVE GREEN
OUR PLANET
EVERY DROP COUNTS
WORLD ENVIRONMENT DAY
RECYCLE
SOCIAL MEDIA POST TEMPLATE
Zero Waste
Reuse & Recycle

Figure 11.94 Collage of design elements that often find application in the sustainability field.

- For medical themes, the design language tends to be clear and straightforward. The colors are often white, blue, or turquoise, as shown in Figure 11.95.

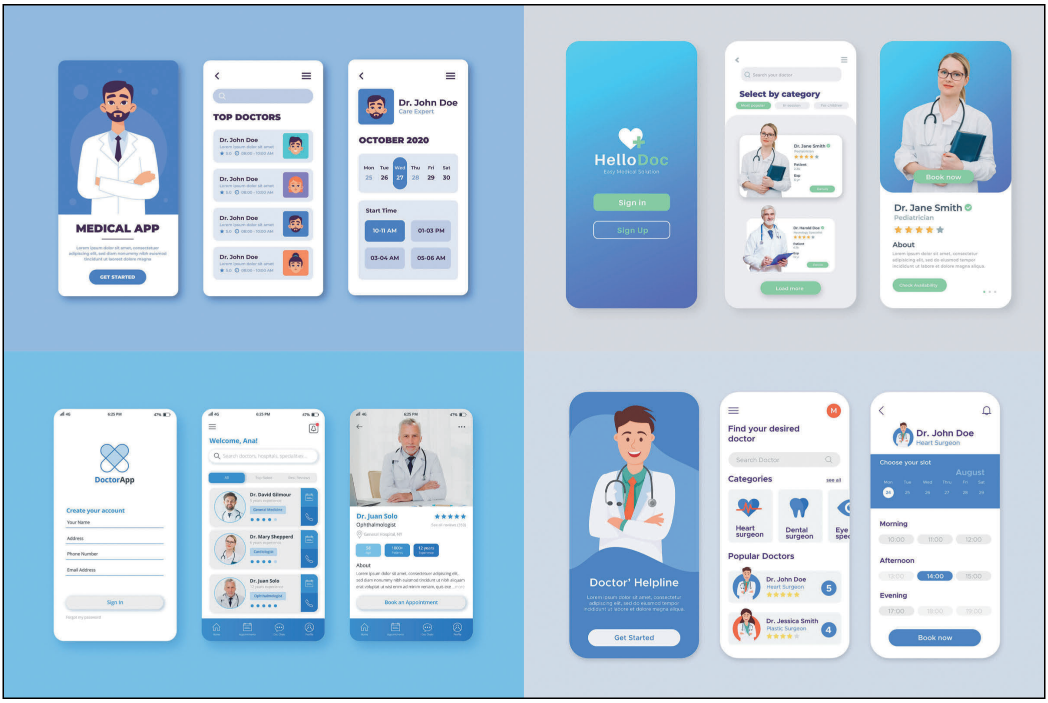

MEDICAL APP
GET STARTED
TOP DOCTORS
Dr. John Doe
Care Expert
OCTOBER 2020
Start Time
10-11 AM
01-03 PM
03-04 AM
05-06 AM
HelloDoc
Easy Medical Solution
Sign In
Sign Up
Select by category
Load more
Book now
Dr. Jane Smith
Pediatrician
About
Check Availability
DoctorApp
Create your account
Your Name
Address
Phone Number
Email Address
Sign In
Welcome, Ana!
Dr. David Gilmour
Dr. Mary Shepperd
Dr. Juan Solo
Dr. Juan Solo
Ophthalmologist
About
Book an Appointment
Doctor' Helpline
Get Started
Find your desired doctor
Search Doctor
Categories
Heart surgeon
Dental surgeon
Popular Doctors
Dr. John Doe
Heart Surgeon
Dr. Jessica Smith
Plastic Surgeon
Dr. John Doe
Heart Surgeon
Choose your slot
August
Morning
Afternoon
Evening
Book now

Figure 11.95 Collage of design elements often used in the medical field.

Especially if you're new to an industry or are developing a color and design language for a company for the first time, a helpful step is to look at the colors of the industry. You can then build on this knowledge to determine the visual direction of your system, product, or service. In doing so, you can focus on these typical industry elements or specifically ignore them.

However, you should not ignore industry cues to such an extent that people find it difficult to associate your work with the industry intuitively! If so, your solution may not even be noticed by interested parties because it does not fit the expected search pattern. Therefore, we recommend developing a feeling for how the industry presents itself.

Current Trends

Just as with general design research, we recommend looking for current trends in visual design with visual design research. Unlike general design research, however, the focus is not on interaction patterns but on the current visual representations that are in vogue.

Every few months or every few years, you'll see new design trends spreading. Some recent trends include changes in shadows, 3D effects, or even color gradients.

Looking at smartphones again, notice how design trends have changed over the years. Figure 11.96 shows the logos of the various generations of iOS, Apple's operating system for the iPhone. Although a change did not occur every year, notice how easily a design can be perceived as outdated or more boring just one year after its launch.

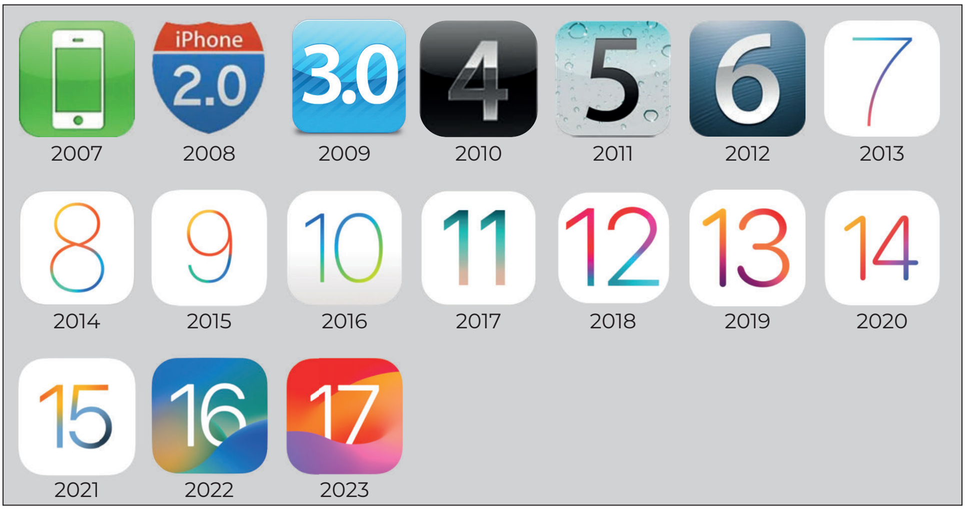


Figure 11.96 Apple iOS logos of the past (https://apple.fandom.com/wiki).

The important task is not necessarily to reflect the current trend, but instead to be aware of current trends and consciously decide for or against following them.

The Style of the Application

Finally, we recommend looking again at the visual style of comparable applications at the lowest level. These applications can be your own company's applications (predecessors or other applications) or the applications of your competitors.

Notice that competing applications are often similar in visual style and interaction. But occasionally, when conducting research, you might come across an idea that is really clever, that is attractively implemented, and where it makes sense to think about integration. Of course, when we say *integration,* we don't mean copying the content exactly but instead adopting the idea to your own company's color and design language.

11.12.2 The Association Space

The actual entry into the visual design takes place with the help of the query of the association space, which we described in detail in Chapter 7, Section 7.4.1. In this section, we'll now focus on determining the desired association space for your solution.

In our experience, people find it extremely challenging to describe a desired design with the associated design elements (e.g., typography). Especially if laypeople are involved in the design, they often lack the words to address and express the target design. Often, they don't get beyond "kind of cool," which really doesn't help the design. But what can usually be defined by stakeholders are the desired associations—and we take advantage of these desired associations.

Our goal is to gather desired associations as defined by our *stakeholders*. Our task as *designers* is to transfer these associations to the solution. Ultimately, the stakeholders or later users should have these associations when they look at the UI. You can read more about the association space test in Chapter 12, Section 12.19. Let's focus on the question: "How do we proceed so that the stakeholders define the desired associations?"

We can use the association space as a query element to capture the whole view in a more structured way. In the simplest form, stakeholders are given adjectives from which they can select the most important three to five words. For this purpose, we have created lists of adjectives that contain terms such as "sustainable," "modern," "traditional," "sporty," or "high-quality." Based on the previous design research, these lists contain a pre-selection of about 20 to 50 emotional adjectives. As a suggestion, you can find a list of almost 200 adjectives at *http://s-prs.co/v577800*.

An alternative is to work entirely without a predefined list. In this case, stakeholders can freely name the adjectives they want to associate with the product. The advantage of this less restrained method is that you can also name adjectives that are not on your list. On the other hand, the disadvantage is that, in the next step, you must merge the stakeholders' responses into three to five characteristics, which can be much more difficult with free naming. With the former variant, the opposite is true: You're more limited and have more preliminary work—but the common consensus is usually easier to determine.

You've already figured it out from the pros and cons: The next step is establishing stakeholder agreement on the three to five adjectives that characterize your solution.

As soon as this task is complete, you know which associations your stakeholders, customers, or users expect from you or, more precisely, from the solution. We usually plot these associations in a network graph and, together with the stakeholders if possible, determine a prioritization. The result of this first phase is a graphic similar to the one shown in Figure 11.97.

With this knowledge, you can start the following steps and offer the first design styles matching the desired client's style (from the visual design research). For this task, select a few points from the defined association space, as shown in Figure 11.98, and create proto-mood boards for these points. You can choose the selection criteria because we're just looking for a starting point. This approach usually performs best when we have defined these points with the stakeholders.

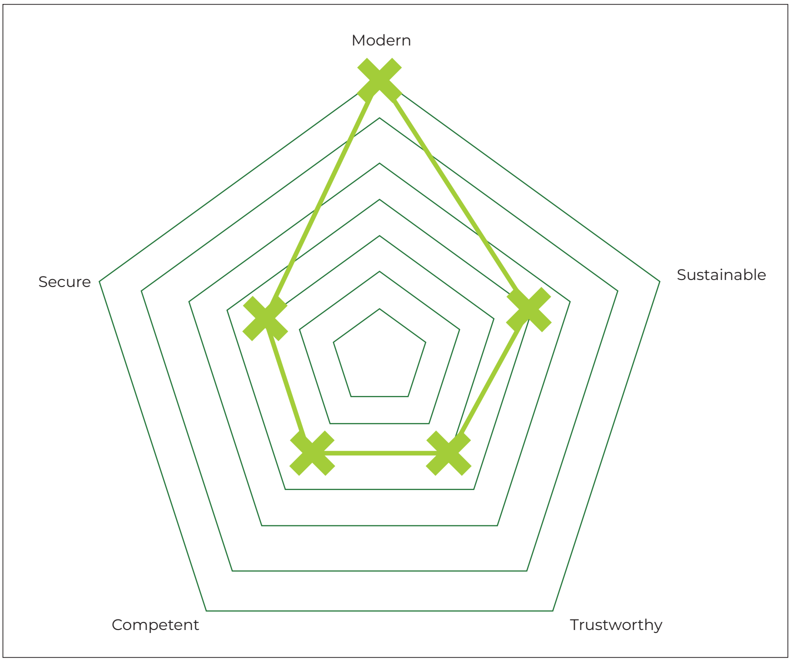


Figure 11.97 Association space (five adjectives) developed together with stakeholders and a prioritization that makes the space smaller and thus more defined.

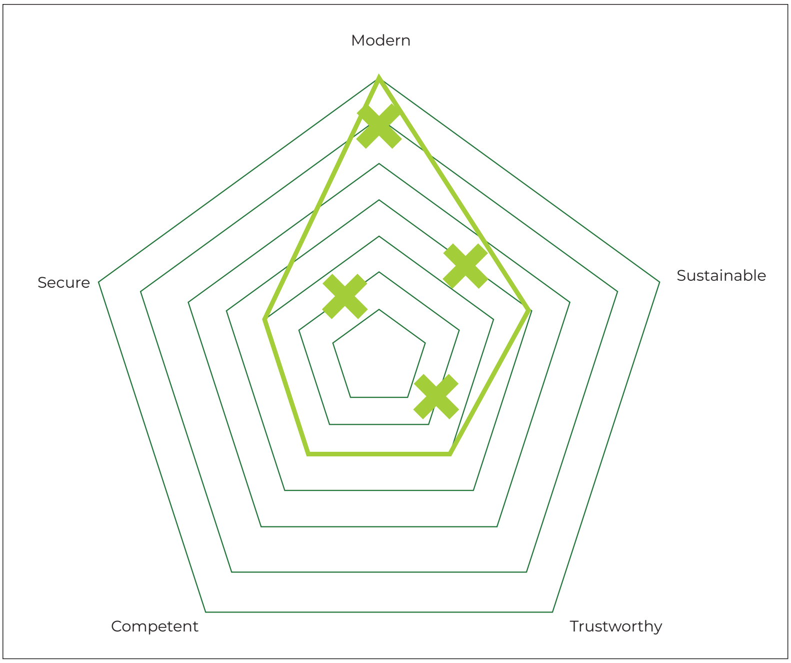


Figure 11.98 Four selected locations in the association space for which proto-mood boards are created in the next step.

11.12.3 Proto-Mood Boards

The next step is to assign the style elements you already have from your style guide to each point. In this step, we want to understand what we already have and can use and what may need to be styled.

We resort to a structured mood board view and use our proto-mood boards to make this alignment. A proto-mood board is a collection of the basic elements in the design in a "set box" look, as shown in Figure 11.99.

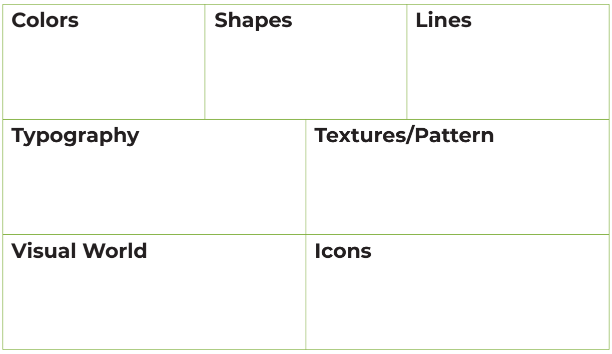


Figure 11.99 Template of our proto-mood boards.

Our proto-mood boards usually contain the following elements:

- Colors (we already have)
- Shapes (e.g., round or square)
- Lines (e.g., thin and filigree or thick and solid)
- Textures/pattern (e.g., "soft as cotton" or "hard as concrete")
- Typography (e.g., curved or straight)
- Visual world (e.g., photos or illustrations)
- Icons (e.g., flat or three-dimensional)
- If necessary, others depending on the application

However, this template is not rigid, as you can see from the last item in this list. Depending on your requirements, areas can be larger or smaller or can be entirely omitted, and new ones can be added. Thus, you can customize—but you should ensure that all basic elements necessary for your solution are included and always use the same template in your project afterward to enable easier comparability.

We can now use this template for each point in the association space we want to look at first and copy the basic elements that already fit. Have you found typography in the very modern style guide? Then, off it goes into the associated proto-mood board! Your color scheme fits well with "sustainable"? Then, put it in the set box placed there! We think you understand the principle.

If sorting is not easy, first use research and your design expertise to select adjectives to define a proto-mood board and thus use them as visual cues.

Gaps regularly occur during the sorting process, for example, if your style guide contains no imagery but you need them for the solution. How we close these gaps is described in the next section.

11.12.4 Closing Gaps in the Proto-Mood Board

To fill in the gaps in proto-mood boards, we usually involve stakeholders again. On the one hand, you could close these gaps with your own design expertise and bring these suggestions into a stakeholder workshop. The second option is to close these gaps directly with stakeholders.

We typically use a combination of these approaches in our projects. We bring reasoned proposals to the stakeholder workshop and then use a tool called a *design compass* for gaps we didn't "hit" with our ideas.

A design compass is just a collection of basic design elements in different styles, as shown in Figure 11.100, that you can create yourself with relatively little effort. With this collection, non-designers, perhaps guided by a design expert, can select basic elements that close the gaps in your proto-mood boards.

Of course, you can also define elements without a design compass. But we decided to use a template in stakeholder workshops because design is subjective. Without any orientation, we noticed that, especially at the beginning, many loops are made in the design before a basic agreement is reached.

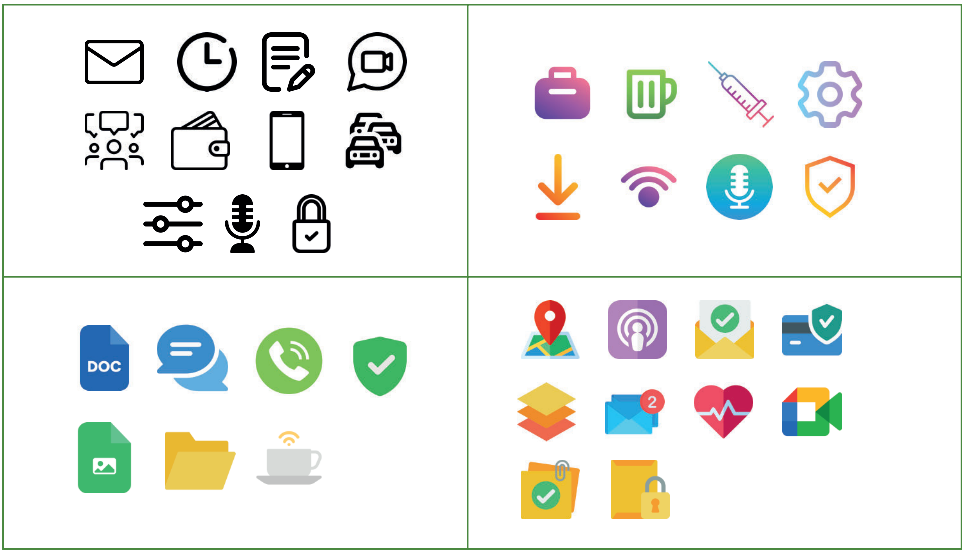


Figure 11.100 A selection of modern icons. Our design compass represents a collection of basic design elements enabling easier selection for users, customers, or stakeholders.

One more note: We don't necessarily use the design compass to hammer out the final design completely but instead use it to find an entry point that serves as a basis for further design by us. We, therefore, take elements picked out in the stakeholder workshop and develop the proto-mood boards further from this information.

11.12.5 From Proto-Mood Boards to Mood Boards

When a proto-mood board has no more gaps and is visually coherent, we can now speak of a *mood board*, as shown in Figure 11.101, which may be a bit of semantic juggling. Still, we have found that discussions with stakeholders are much easier if the name already suggests whether a proposal is "prototypical" or, in our view, finished.

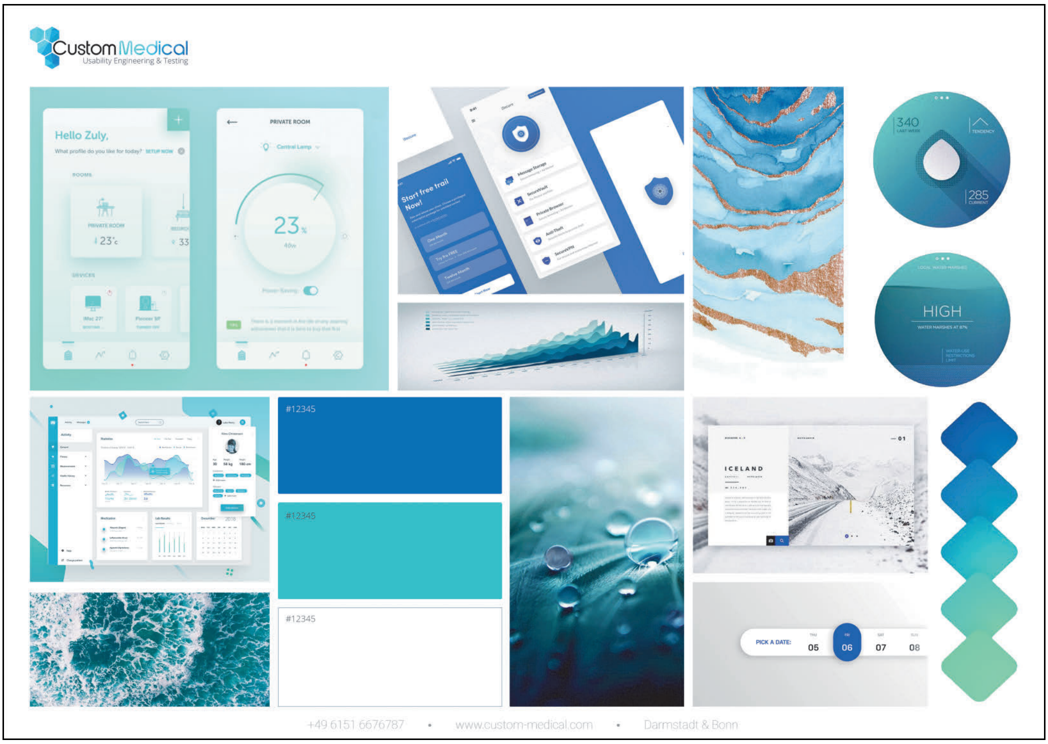

Custom Medical
Usability Engineering & Testing
Hello Zuly,
PRIVATE ROOM
23
Start free trail Now!
340
285
HIGH
#12345
#12345
#12345
ICELAND
PICK A DATE:
05
06
07
08
+49 6151 6676787 • www.custom-medical.com • Darmstadt & Bonn

Figure 11.101 Example mood board for Healthyfy.

Once mood boards have been created for the chosen points in the association space, the next step is to select them. A best practice is again involving all stakeholders in the selection process. In this case, our experience shows that the broader the contribution to a design concept, the better it is anchored in the company.

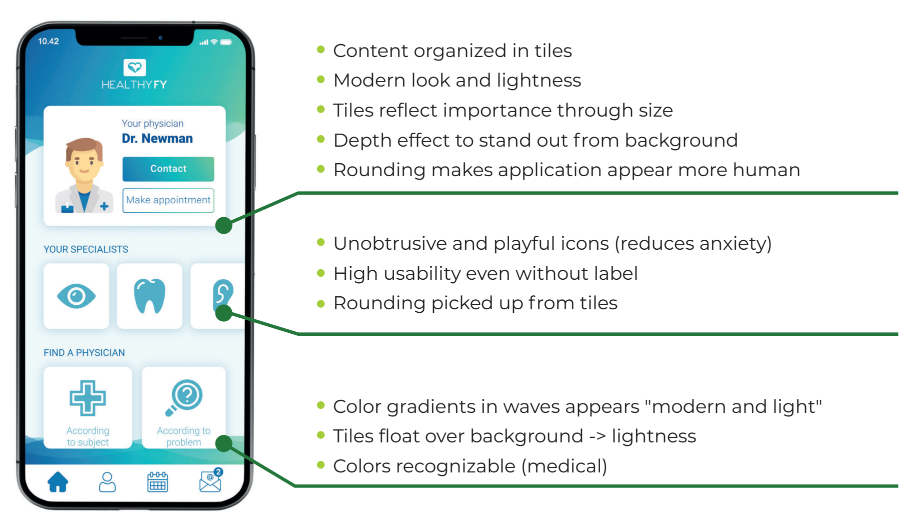
At the end of the selection process, a mood board describes the solution to be designed.

11.12.6 The Visual Design Concept

At this point in the visual design, we have a selected mood board that gives a first impression of the effect of the later visual design. In this step, we'll now use the mood board as a basis for visually designing the first central control elements and some views, thus further defining the visual style of the solution. The central questions we ask now include "How do we want to present individual content, and how do the individual visual elements interact with each other?" Our aim here is to create a *reasoned style* that is made up of individual UI elements.

Therefore, we now look at the *wireframes*. When a visual design comes into focus, these should already be mainly in place and ideally have been tested several times with users—making them the perfect basis for designing the individual elements. Now, we apply the developed visual basis to our wireframes: Shapes get the derived shapes, colors are used according to the selected color world, and so on.

We document each decision and rationale so that a collection of designed elements with their associated rationales gradually emerges—our visual design concept. An example page from such a concept for Healthyfy is shown in Figure 11.102.



- Content organized in tiles
- Modern look and lightness
- Tiles reflect importance through size
- Depth effect to stand out from background
- Rounding makes application appear more human

- Unobtrusive and playful icons (reduces anxiety)
- High usability even without label
- Rounding picked up from tiles

- Color gradients in waves appears "modern and light"
- Tiles float over background -> lightness
- Colors recognizable (medical)

Figure 11.102 Excerpt from a visual design concept for Healthyfy.

11.12.7 Creating Visual Prototypes (Click Dummies)

Once the basic visual style of the application has been derived and implemented on some initial screens, the next step is to gather stakeholder feedback. As with the wireframes, we create click dummies or further develop the existing wireframe click dummies by integrating the visual design.

Usually, we won't put all possible workflows into a visual prototype but instead will limit ourselves to the most essential processes. We determine the importance of workflows according to the purpose of the click dummy:

- If conducting a test with users, then mainly critical workflows and very frequently used workflows are implemented.
- Regarding "internal marketing," we primarily implement the workflows most relevant to the relevant decision-makers and enable a good understanding of the application.

We had already taken the same approach with the wireframe click dummies (Section 11.9). Thus, in our case, click dummies with the most important workflows already exist. Nevertheless, since implementing a visual prototype is costly, we use them selectively. In short projects with smaller applications, a complete visual prototype can replace a style guide as documentation. For extensive applications, a separate style guide is more advisable. We'll look at this topic more closely in the next section.

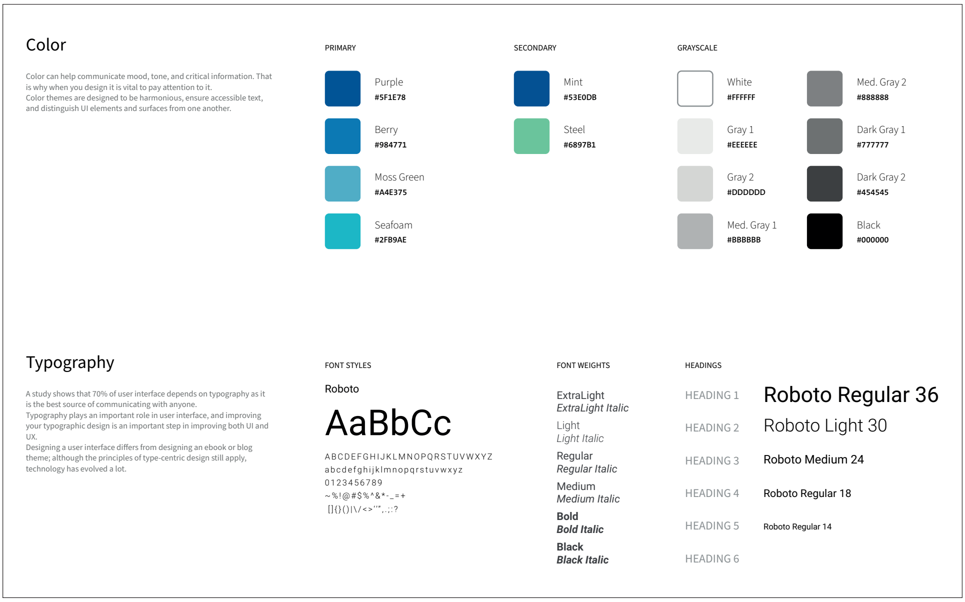
11.12.8 The Style Guide

The final stage in visual design is the documentation of all design decisions made concerning the product and, thus, the definition of the future design in the form of a style guide. Style guides can either be developed as a single document or provided as a digital resource based on the prototypes.

When created as a document, a style guide is usually a PDF file that documents the application's colors, fonts, and interface elements. Usually, we build on a visual design concept (Section 11.12.6). However, this concept is supplemented by a library of all possible UI elements and their designs. Particular states of the solution, such as warnings, are also documented in the style guide, as shown in Figure 11.103.

When creating style guides, you should ensure no contradictions exist with existing style guides or within your style guide. Working with style guide documents is helpful in implementation but can lead to frequent scrolling, especially in extensive projects, because the information is often scattered throughout the document. For example, usually you'll have a section on fonts, one on colors, one on interaction elements, and so on. But if a button must be designed in the implementation, it contains a color, a font, and the button itself. To deal with this problem in the documentation, duplication of style guide content might be necessary, which leads to a higher maintenance effort.

One way to avoid this issue is to use a digital style guide based on your click dummy. In your click dummy, you have already defined most elements and can document them more easily. If you later want to design a new area in the UI, you can copy the elements from the digital style guide. One benefit is that you can work much more consistently and quickly. For example, you can always display element properties directly and thus see the font without switching to the typography area first.



Color

Color can help communicate mood, tone, and critical information. That is why when you design it is vital to pay attention to it.
Color themes are designed to be harmonious, ensure accessible text, and distinguish UI elements and surfaces from one another.

PRIMARY

Name	Code
Purple	#5F1E78
Berry	#984771
Moss Green	#A4E375
Seafoam	#2FB9AE

SECONDARY

Name	Code
Mint	#53E0DB
Steel	#6897B1

GRAYSCALE

Name	Code	Name	Code
White	#FFFFFF	Med. Gray 2	#888888
Gray 1	#EEEEEE	Dark Gray 1	#777777
Gray 2	#DDDDDD	Dark Gray 2	#454545
Med. Gray 1	#BBBBBB	Black	#000000

Typography

A study shows that 70% of user interface depends on typography as it is the best source of communicating with anyone.
Typography plays an important role in user interface, and improving your typographic design is an important step in improving both UI and UX.
Designing a user interface differs from designing an ebook or blog theme; although the principles of type-centric design still apply, technology has evolved a lot.

FONT STYLES

Roboto

AaBbCc

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
~%!@#$%^&*-_=+
[]{}()|\/<>‘’”,.;:?

FONT WEIGHTS

ExtraLight
ExtraLight Italic
Light
Light Italic
Regular
Regular Italic
Medium
Medium Italic
Bold
Bold Italic
Black
Black Italic

HEADINGS

Heading	Style
HEADING 1	Roboto Regular 36
HEADING 2	Roboto Light 30
HEADING 3	Roboto Medium 24
HEADING 4	Roboto Regular 18
HEADING 5	Roboto Regular 14
HEADING 6	

Figure 11.103 Excerpt from the Healthyfy style guide.

Depending on the tool, you might be able to export properties directly as a CSS file so that the design is more or less directly transferred into your own (web) applications.

11.13 Excursion: Low-Code and No-Code

At this point, let's take a brief excursion into an area that has become increasingly important in recent years—*low-code* or *no-code platforms*. These platforms aim to simplify the creation of applications without much programming effort.

The low-code approach is nothing new. In the 1990s, some applications enabled the development of applications with low programming effort. A probably well-known example of these early applications is Visual Basic.

The idea behind low-code applications is to develop an application without (or at least not predominantly) using a programming language or using code but simply by configuring graphical interfaces. This endeavor made it easier for non-programmers to create applications. For some companies, one advantage was that subject matter experts could be involved much more directly in developing an application and thus show what they expect or need from it. In the best case, business experts could even build their own applications.

As the name *low-code* implies, the platform makes it easier to create applications, but it is still impossible to do so entirely without programming. When low-code platforms are used, around 80% of the development of an application can be carried out via graphical UIs. In contrast, the remaining 20% requires knowledge of programming.

In contrast, *no-code platforms* promise to implement applications entirely without programming, creating complete applications without any code written by the user. Ultimately, no-code platforms do not offer any ability to introduce one's own code into the development. As a result, no-code solutions today quickly reach their limits—simply because it is virtually impossible to connect complex systems or incorporate special requests.

We wanted to mention the topic of low-code/no-code development in this section to show you an alternative to classic development work. If you familiarize yourself with low-code platforms, you can make an excellent and almost seamless transition from prototype to finished solution without much additional implementation effort. One prerequisite is that you strictly follow the capabilities of the low-code or no-code platform in your design because adjustments later on are either costly or simply not possible.

11.14 Example: Possible Design Steps for moveHUB

We meet moveHUB again when the three founders are creating the first designs. For the first workshop, they picked the main problem of the persona *Sebastian "The Student" Singer*—not being late for class. As a reminder, here is the matching user story again:

> *As Sebastian "The Student" Singer, I want to arrive at the lecture hall on time to not sit alone in a large lecture hall or be late in front of everyone.*
>
> *The current inability to do this is causing that I sometimes fritter away time (rarely), but I usually miss a lecture because I'm embarrassed to be late in front of everyone.*
>
> *Enabling Sebastian to do this would allow him to hear the complete study material in lectures, reducing his stress significantly.*

For the first workshop, the three founders chose the creative method of *brainwriting*. They have decided that everyone will first be given 5 minutes to write, draw, and/or scribble as many solutions as possible on paper. Afterward, solution ideas will be presented, whereby only comprehension questions are allowed. Next, the team works to improve each solution together and then comes to an evaluation.

When the timer starts with the 5 minutes, everyone immediately starts putting their ideas on paper, as shown in Figure 11.104. The 5 minutes are up faster than expected, and the team adds another 5. This time also passes quickly. Nevertheless, they decide to go into the round of introductions now.

The three founders find it difficult to adhere to their rules. The temptation to directly express one's opinion on each idea is too strong. But since opinions at this point would create an unhealthy dynamic and the choice of words tends toward "my solution" and "your solution," Emma throws in the towel. She suggests that everyone take a break and then return to the table, where the rules must be followed. If someone doesn't, they have to pay for dinner. It's a plan that works for the three of them. Together, after a half-hour break, they manage to present ideas to each other and then improve upon them.

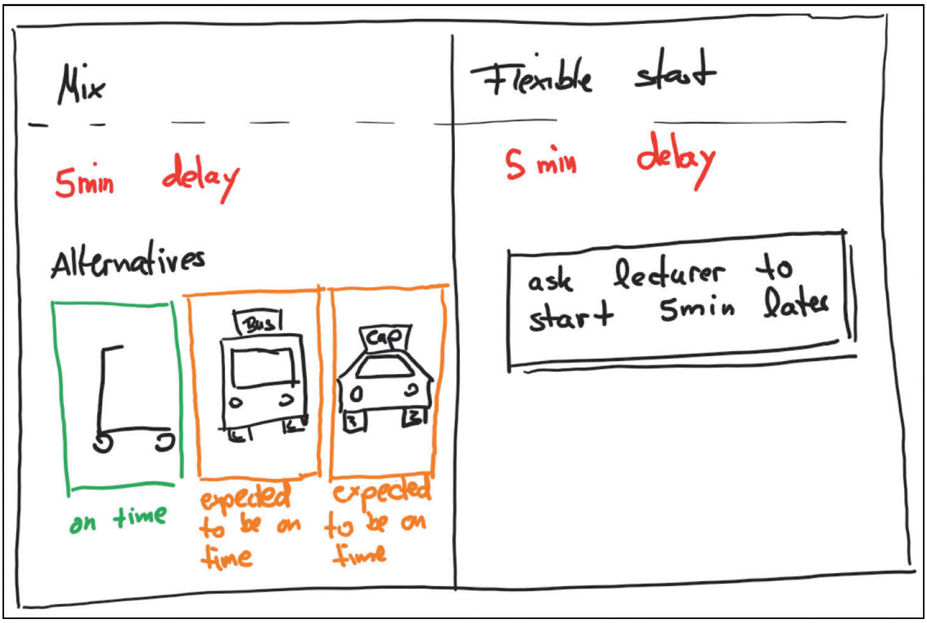

Mix
5min delay
Alternatives
Bus
Cap
on time
expected to be on time
expected to be on time
Flexible start
5 min delay
ask lecturer to start 5min later

Figure 11.104 Excerpt from the moveHUB brainwriting session results: "mix of all modes of transport" (left) and "flexible start of lectures" (right).

For the idea "mix of all means of transport," many suggestions are considered for improvement and further development. The following points are mentioned, among others:

- In addition to public transport such as buses and trains, rental vehicles (e.g., electric scooters) could also be included. If the app knew where these available vehicles were located, what the traffic situation was like, and how the public transportation system operated, then the app could accurately suggest which means of transportation would reach the destination.
- Users should be able to set which mode of transportation they are willing to use. For Sebastian, as a student, a cab is probably not an option. For a working Sebastian (maybe, in hindsight, they should separate the persona?), perhaps cabs are fine.
- The app should have access to Sebastian's calendar. Alternatively, Sebastian should be able to select which lecture he is listening to. Then, the app could tell Sebastian where he has to be without his intervention.

The team is enthusiastic. Ideas were discovered and then improved upon. The evaluation was also so easy that the three decided to add a few more hours and go through all the user stories in a similar fashion.

The three founders have succeeded. For them, it's clear that they've found a suitable method, and they can start wireframing next. But this step will have to wait until tomorrow because Matthew has to pay for dinner tonight.

11.15 Example: Possible Design Steps for BusinessBooster

The last time we assisted Markus Meier's team, we helped define the requirements list. In this process, it became clear that further user research was needed to understand the reporting functions desired by customers more precisely. This user research revealed, among other things, that a graphical evaluation is desirable. Furthermore, the reports should be adaptable to the customer's own needs.

Meanwhile, the team spent several iteration loops in the data-driven UX design process, repeatedly designing wireframes, gathering user feedback to sharpen requirements and the context of use, and then starting again with a better solution.

Let's look at the current status of BusinessBooster. Markus's team has discovered that the targeted analytics module should have two interfaces:

- An administrative interface for creating reports
- An interface for the primary users

The administrative team conceived the BusinessBooster interface more technically: With the help of a scripting language, reports can be defined entirely and freely at the user's discretion. The most recent feedback was that such an interface was suitable in theory but too complex. Only a few customers liked learning a scripting language, but they all found the possibilities impressive.

The administrative view has been revised in the current design solution, as shown in Figure 11.105.

Now, users can access ready-made report modules that can be placed by dragging and dropping. In this case, the desired data can be inserted via parameters. Scripting options have also been preserved, but now, a code block can also be added. Subsequently, this code block element can be freely programmed. In this way, the team hopes to have found a happy medium: Much of what is often needed is available "ready-made," and for everything else, you can use the code block. Markus even senses further business: "Wouldn't it be a good idea to offer more complex report modules for money? Alternatively, we could also offer the programming of your reports for a fee! I wonder what the users think about this."

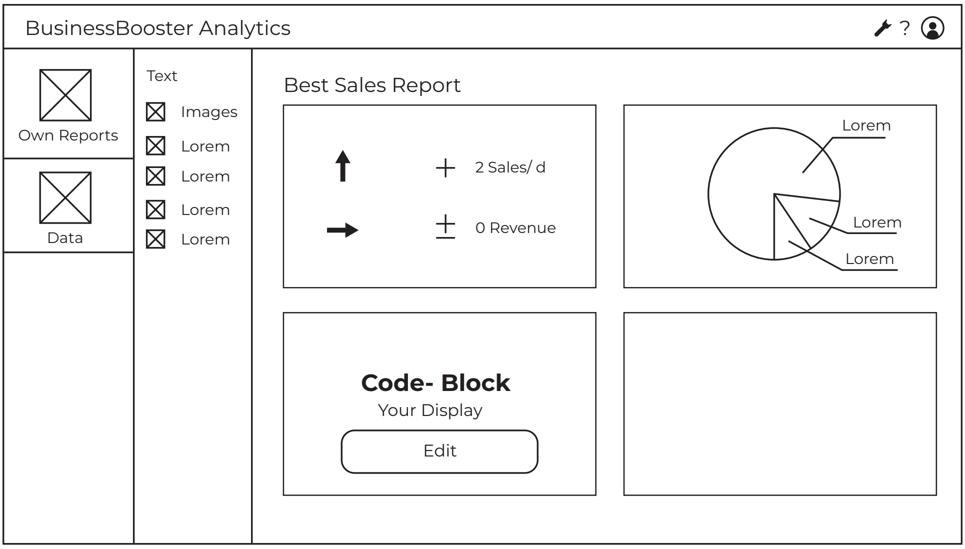


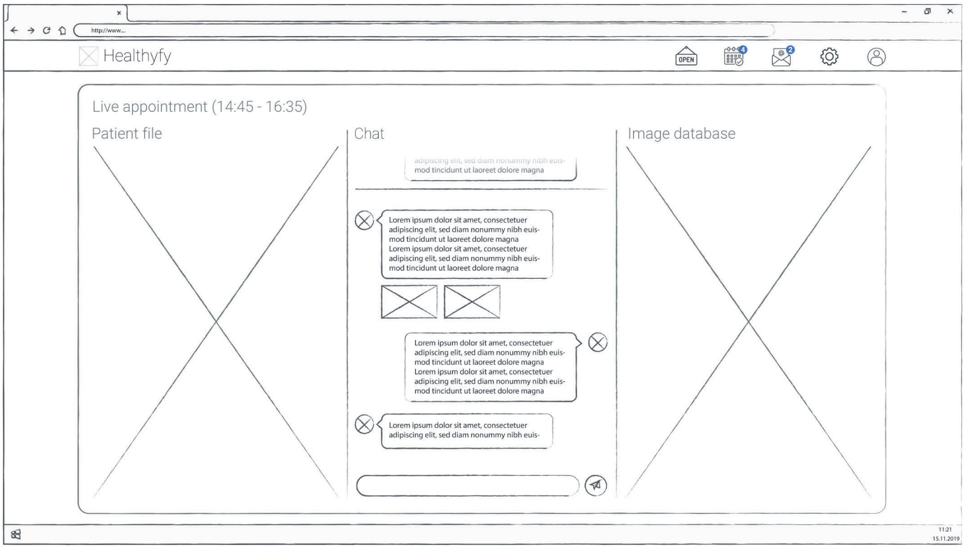
Figure 11.105 Wireframe for the analytics module for BusinessBooster. You can create your reports using predefined elements, for instance, code blocks.

11.16 Example: Possible Design Steps for Healthyfy

Our last contact with Healthyfy was some time ago. Linda and her team have accomplished quite a bit in the meantime. Various iterations have already been run, and in principle, the wireframes are ready now, as shown in Figure 11.106. Some gaps still exist, but from the team's point of view, these gaps can be filled in through hard work and no longer stand in the way of the product itself.

Because the wireframes are advanced, the team decides to continue on two tracks. On the one hand, the wireframes are finalized and tested with users. On the other hand, user tests are also conducted to test the visual design. The idea is that, if all wireframe questions are checked off first in each user test, and then the visual design is discussed for 10 minutes, the team can save testing time and speed up development.

The team has already done a lot of work on this design. Both the brand and the basic look have already been defined. Right now, the team is working on the elaboration of various areas. One question is whether the central chat area—an area proven essential for patients and doctors—should be visually highlighted to further emphasize its importance. Therefore, two variants were created, which will be shown to users in the next step, as shown in Figure 11.107.

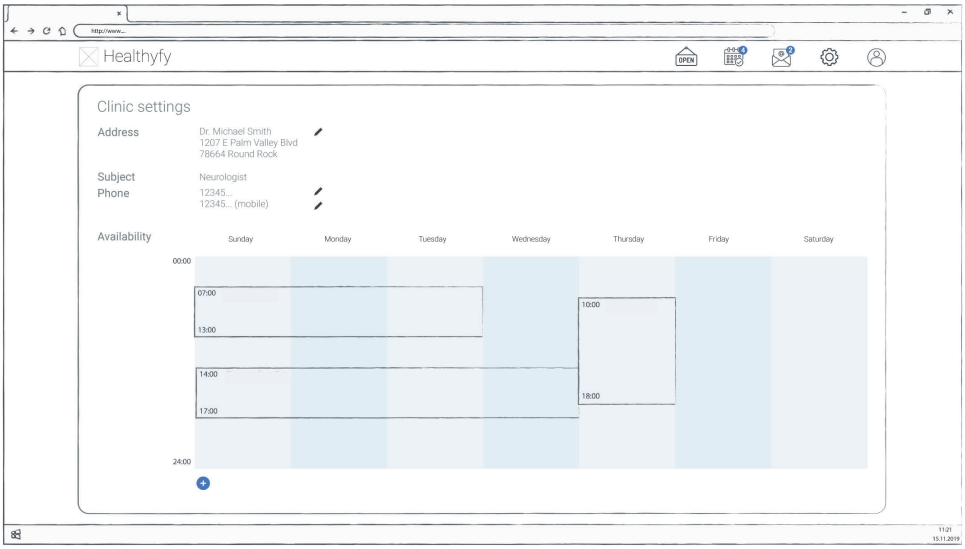

http://www...
Healthyfy
OPEN
4
2
Live appointment (14:45 - 16:35)
Patient file
Chat
Image database
11:21
15.11.2019

adipiscing elit, sed diam nonummy nibh euis-
mod tincidunt ut laoreet dolore magna

Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euis-
mod tincidunt ut laoreet dolore magna
Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euis-
mod tincidunt ut laoreet dolore magna

Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euis-
mod tincidunt ut laoreet dolore magna
Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euis-
mod tincidunt ut laoreet dolore magna

Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euis-


http://www...
Healthyfy
OPEN
4
2
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
00:00
07:00
13:00
14:00
17:00
10:00
18:00
24:00
+
11:21
15.11.2019

Clinic settings

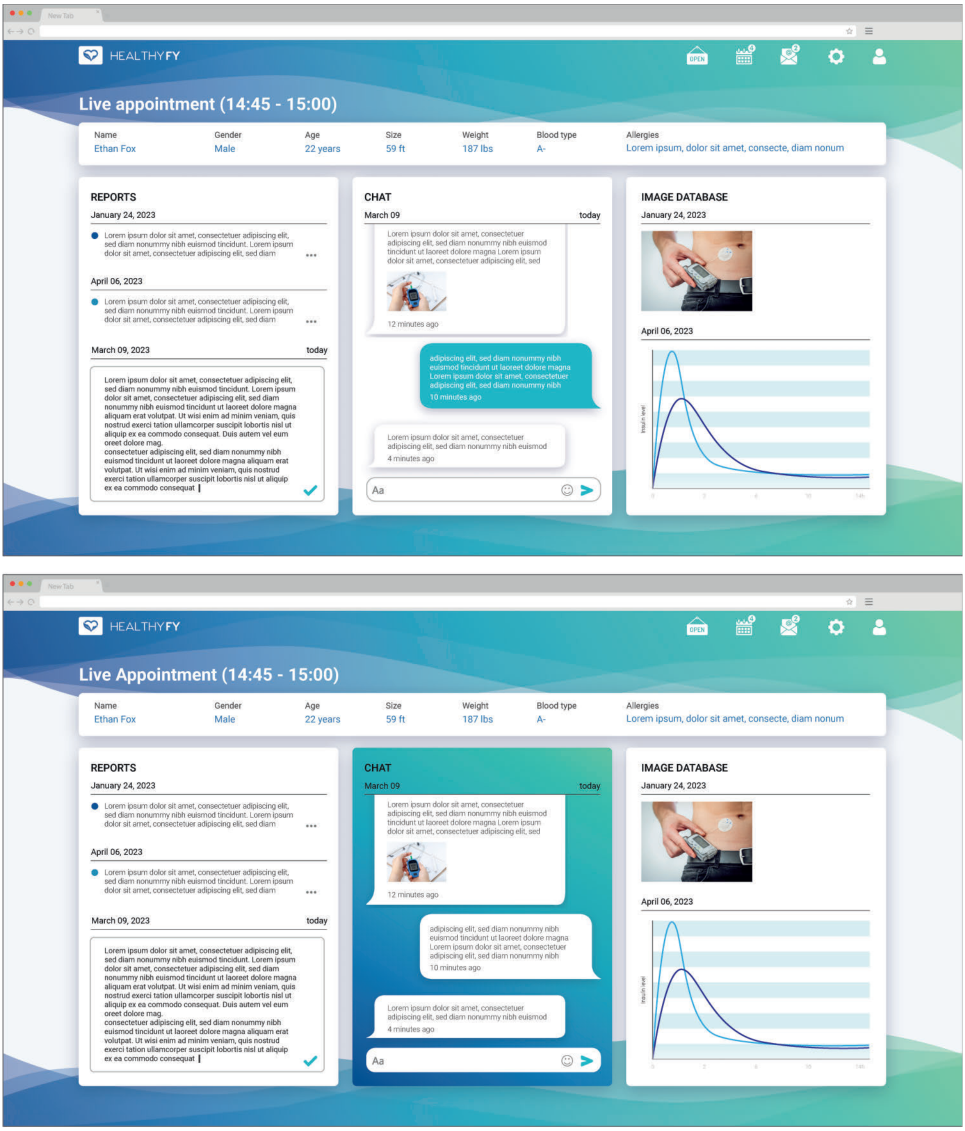
Address
Dr. Michael Smith
1207 E Palm Valley Blvd
78664 Round Rock

Subject
Neurologist

Phone
12345...
12345... (mobile)

Availability

Figure 11.106 Healthyfy's wireframes: Most areas are defined and work in the envisioned use cases.



Live appointment (14:45 - 15:00)

Name	Gender	Age	Size	Weight	Blood type	Allergies
Ethan Fox	Male	22 years	59 ft	187 lbs	A-	Lorem ipsum, dolor sit amet, consecte, diam nonum

REPORTS

January 24, 2023

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam ...

April 06, 2023

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam ...

March 09, 2023 today

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum oreet dolore mag.
consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat ✓

CHAT

March 09 today

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed
12 minutes ago

adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh
10 minutes ago

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
4 minutes ago

Aa

IMAGE DATABASE

January 24, 2023

April 06, 2023

Live Appointment (14:45 - 15:00)

Name	Gender	Age	Size	Weight	Blood type	Allergies
Ethan Fox	Male	22 years	59 ft	187 lbs	A-	Lorem ipsum, dolor sit amet, consecte, diam nonum

REPORTS

January 24, 2023

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam ...

April 06, 2023

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam ...

March 09, 2023 today

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum oreet dolore mag.
consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat ✓

CHAT

March 09 today

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed
12 minutes ago

adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh
10 minutes ago

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
4 minutes ago

Aa

IMAGE DATABASE

January 24, 2023

April 06, 2023

Figure 11.107 Two variants of the central chat from the physician's view at Healthyfy. Which one will users choose?

Chapter 12

Data-Driven UX Design in Detail: Evaluating Design Solutions

"You're going to make mistakes. The key is to learn from them as fast as possible and make changes as soon as you can."
—Tiger Woods (one of the most successful golfers in history)

Now it's here: the moment of truth. You've analyzed the context of use, derived usage requirements, and designed for all you're worth. In this step of the data-driven user experience (UX) design (3DUX) process, you find out if all the work up to this point has been worth it.

Even though evaluating design solutions is the first step where exiting from the development cycle is possible, if your project is like most projects, evaluation is not the final step but a prelude to improving everything that has gone into the project so far.

Therefore, we encourage you to consider evaluation as an intermediate step and consider your product "permanently in development." This idea is especially true since all the test methods mentioned in this chapter can, of course, not only be carried out with final products but bring about the most significant benefit if you test with prototypes early on in development. A special case of testing is called *summative evaluation*, which takes place *after* development is complete. We'll discuss when this special case makes sense in testing, and when it doesn't, in the next section.

12.1 Formative versus Summative Testing

All human-centered development processes in this book are based on the shared conviction that design should be tested. Behind this belief is the view that only the actual users of a solution can decide what makes a system, product, or service good.

The better and more extensive the user research in the context of use analysis was, and the more familiar user researchers are with the setting, the closer you probably are to a good solution with your system, product, or service, even in the first attempt. The more innovative, new, and unknown the users, tasks, and settings are, the further away you'll probably be with your first solution proposal. This statement is not a bad thing. It is precisely this discrepancy between where you are after design and where you want to be, based on your goals/key performance indicators (KPIs; see Chapter 3), that the formative evaluation of design solutions is all about.

If you aim to reduce this difference between the target design and the actual design and improve your solution, then we speak of a *formative evaluation* (Döring & Bortz, 2016). Simply put, we could say that a formative evaluation helps you determine if you're on the right track with your development, as shown in Figure 12.1.

Figure 12.1 What is the right path for your solution? Formative studies tell you.

Accordingly, a formative evaluation can also be anything that helps you answer that question. No set requirements exist for formative evaluations except that they help you understand how good your solution currently is and what you would need to improve it. From our experience, this assessment is hard enough because the quality of answers you get is directly related to the quality of questions you ask.

In this chapter, we'll provide you with a set of "good questions" that you can use to assess the status of your solution. Formative evaluations are usually conducted in collaboration with the development team. In this context, a strong interest exists on the part of stakeholders to witness every step and observe every test. After all, it's "their baby," and they are concerned about what will happen to it.

On the other hand, a *summative evaluation* is an assessment carried out with the finished system, product, or service (Döring & Bortz, 2016). In this case, the focus is no longer on optimizing the solution. When you start a summative evaluation, you want to understand how well you have mastered everything, as shown in Figure 12.2.

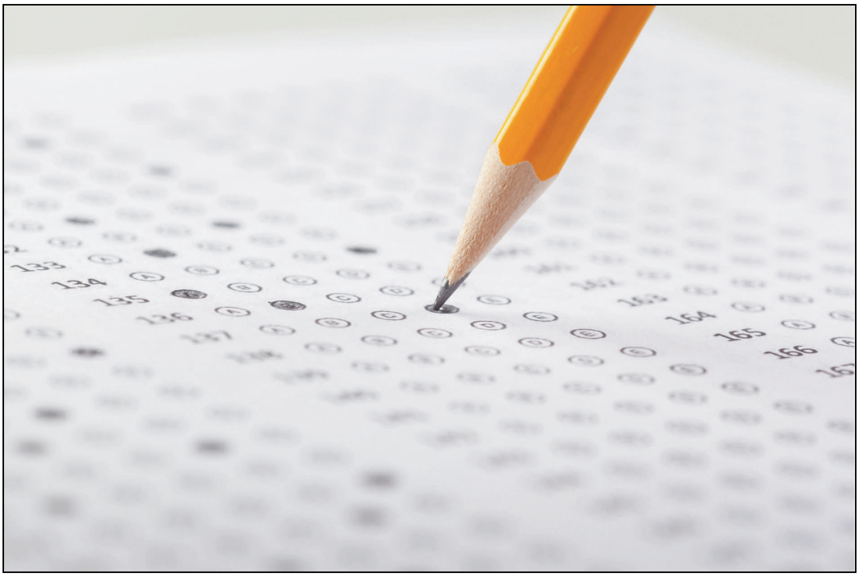


Figure 12.2 A summative evaluation is like a final exam in school. Afterward, you know how well you did with your solution.

Outside the medical sector (which we discuss separately in Section 12.24), summative evaluations occur less frequently and are seldom (at least in our experience) initiated by a development team. Summative evaluations are determinations of the current state, and therefore, two general groups are interested in conducting these evaluations:

- The marketing department
- Independent testing institutions

When summative evaluations are commissioned in a company, this evaluation is usually performed by the marketing department. This choice always happens when a company feels it has developed something great and wants to show it off. You're probably familiar with the text on shampoo bottles that says, "95% of the women in our study recommend this shampoo to a friend. (Surveyed in a study of 30 female participants.) These typical results of a summative study are prompted by a company's marketing. The goal is to show how good a product is and make that result available to the general public. In rare cases, a company's management may also commission a summative study. These studies are usually set up as comparative studies with competitor products to get a feeling for the quality and perception of one's work and to derive any urgent need for action.

The other group consists of *independent testing institutions*. You're probably familiar with *Consumer Reports*. This testing laboratory compares different products without being involved in their development. The advantage of such tests is that independent testing takes place, which gives third parties the certainty that the results have been obtained reliably. Comparative tests are often carried out in these labs, placing several products side by side and assessing them using the same evaluation scheme. However, the usability and UX of these products tend to play a subordinate role in most comparison tests of this kind; the focus is more on technical features.

Perhaps surprising, an increasingly large group of independent testing institutions—which also emphasize usability and UX—are *influencers* who, for example, present and review products as part of their YouTube channels, as shown in Figure 12.3. Many influencers deal intensively with the products and produce high-quality reviews. However, what is true for successful YouTube channels, such as *Unbox Therapy*, is that they influence purchasing decisions with their vast reach.

One point that should be considered when testing through influencers is the question of their independence. Some influencers are paid by the manufacturers of the products; some buy the products to be tested independently and finance themselves through advertising or other sponsors. Especially in the case of influencers who are paid by manufacturers, the marketing of the products can have a more substantial weighting than an objective test. We recommend keeping this fact in mind when looking at influencers' tests.

But whether it's a product you've been handed or bought yourself, in this kind of summative test, an evaluation of the system, product, or service is conducted after development. This type of testing is especially valuable when dealing with an influencer who specializes in a particular group of products (e.g., home automation devices) and uses them daily.

Figure 12.3 Comparison of three smartphones in a video from "Unbox Therapy." (Source: https://www.youtube.com/watch?v=3sZex8SuFb0)

Summative tests in marketing or by independent testing institutions aim to make the quality of the finished solution tangible—regardless of whether the product is good or bad. The testers hang a label on the product, so to speak, and say, "That's great!" Or perhaps, "Well, it wasn't that good after all." This type of evaluation is helpful for anyone who has to decide for or against a particular product.

So, you see, *formative evaluations* and *summative evaluations* do differ, and their main differences can be summarized in the following way:

- Testing for improvement (formative) versus testing for final evaluation (summative)
- Initiated internally by the development department (formative) versus initiated by marketing (summative)
- The user benefits indirectly through a better product (formative) versus the user benefits directly through transparency of quality (summative)

In more and more cases, however, both evaluation forms are also merging. The more agile the development of products becomes, the more frequently users are confronted with an interim development status, which is then permanently improved based on user feedback. This scenario is where the much-cited minimum viable product (MVP) comes into play (Ries, 2011). For a reminder, see Chapter 4, Section 4.3.

When implementing an MVP, the aim is to reduce the functional scope of the initial product solution to the essentials and only expand it over time. However, the initial functions should be smoothly usable in terms of technology, usability, and UX. Through the *build-measure-learn* cycle, "real" MVPs are, therefore, actually data-driven by nature. So, you see, data-driven UX design and MVP are not mutually exclusive concepts but can play together. Every new MVP starts a cycle again.

Still, the methods to evaluate agile products differ from those used to test a washing machine, for example. The more digital—and the less risky—your product is, the further you can venture into the realm of live measurements. We'll describe some methods for these measurements later in this chapter as well. But for now, let's look at what you should be testing in the first place.

12.2 Identifying Knowledge Gaps

If you think back to the beginning of this book, we discussed assumptions versus knowledge at length (see Chapter 6, Section 6.1). The *context of use analysis* is one way to check your assumptions for correctness. The second way to test your assumptions' effectiveness is to evaluate design solutions.

Assumptions can take place at different levels. The top level would be, "The user/customer needs a product like this." At this point, you can re-examine whether the problem exists and whether your solution approach would make a valuable contribution to fixing the problem. That determination mainly happens in the very first iterations.

Later, you check assumptions about the needed features, about the best workflows, and about the appropriate presentation. Any formative tests you do must be designed to show you if you're on the right track with your development. So, when designing your evaluations, you should do so in such a way that the formative tests show you whether your assumptions are wrong.

This goal is more difficult than you might think at first. Because most people—including us, of course—have a problem with being shown that they are wrong. Thus, evaluations and tests are often intuitively created only to show that "everything already fits." Our tip: You'll benefit most from the data-driven UX design approach if you go into evaluations to show that you're wrong. If you fail to prove you're wrong, then maybe you're right.

In addition, you should try to identify things you didn't already know. Formative evaluations should help you understand and learn. The best way to do learn is to look at the big picture, which, in our view, includes objective and subjective data that you can combine to form a whole.

Recall from Chapter 5, Section 5.4, where we briefly described how *objective* and *subjective data* differ and when you're likely to have which type of data available. To summarize briefly once again:

- You need objective data (e.g., from observations or digital usage data) to assess whether users can handle your solution, use it without errors, and achieve the desired goals.
- You need subjective data (e.g., opinions in interviews, scale ratings in questionnaires) to assess whether users accept your solution and are willing to integrate it into their everyday lives.

Therefore, both data types have different purposes, but from our point of view, they can only be interpreted meaningfully together. The prospects for your solution are not good if the solution is entirely error free but no one wants to use it. Nor do you want everyone to be enthusiastic about an idea and the design, but not understand how to use the solution.

For us, you have conducted a successful formative evaluation when you better understand the following information:

- Whether and how well your solution works for the users
- Why this success is the case
- What you need to do to solve the problems found

In most cases, we start with the users when obtaining data. In the next section, however, let's briefly introduce an alternative: using experts to evaluate your design solution.

12.3 User Evaluation versus Expert Evaluation

Evaluation, in simple terms, means what it says. Different people can carry out evaluations. In the field of UX, the two most important groups are users and experts. We've already discussed users in many places throughout this book and discussed experts in a few places. However, both groups are capable of evaluating individual solutions.

In this chapter, we'll introduce you to various methods that you can use to evaluate the quality of your prototypes or your solutions. In this short section, however, we want to raise your awareness of what user and expert evaluations can do. Both have different strengths and weaknesses and can be used in complementary ways. In the best case, you can use both types of evaluations together.

Expert evaluations have an advantage in that they can be used at an early stage and often work across the board. Thus, if your system consists of more than one component and perhaps includes a web application and an app, then an early expert evaluation can help. Depending on the expertise your expert brings to the table, different focus areas are possible in the evaluation:

- If you have an *expert with a technical focus*, that expertise will help you understand how the two systems need to mesh, how data transfers work, and how to enable logging on to both applications with the same credentials.
- If you have a *subject matter expert*, this expert can check whether the correct technical terms have been used and suitable values have been preset.
- If you have a *UX expert*, this expert can assess whether user guidance is implemented sensibly, whether the layout fits together across the different designs, and whether interaction elements are used consistently.

In each case, you'll receive valuable feedback that significantly improves your solution. Best of all, you can usually work with experts very early when your solution is not much more than an initial thought or concept.

Now, you may ask yourself, "If expert reviews are so valuable, why don't I use them exclusively?" In fact, many companies do, but often, the moment a solution optimized solely by expert evaluations hits the market is when the company realizes it's not working. This problem arises because the exclusive evaluation by experts completely ignores that the decisive point for the quality of the solution is not necessarily the technical implementation or the design fit of individual elements to each other. In most cases, a solution's "victory or defeat" is decided exclusively by whether the solution solves a user problem (well enough). And to answer this question, you need users.

Usually, we would present you with many different options that you can use and say, “Everything is okay. Do it the way that works for you.” At this point, however, we want to deviate from that approach because whether you choose to use *user evaluations* or not is so fundamentally crucial to the success of your solution. To put it bluntly, *if you launch your solution and you’ve never talked about it with a user before and never tested it with users, then you’ve done something fundamentally wrong!*

Expert evaluations can always be used in addition to user evaluations, but they must never be the only type of evaluation if you want to ensure that you have found an understandable, exciting, and safe solution.

12.4 Remote versus On-Site

Whether you’re working with experts, users, or both groups, you’ll always decide whether to conduct assessments remotely or together on-site. Recent years, in particular, have significantly boosted the use of home office and digital solutions, so you can rightly say, “Remote works great, doesn’t it!” We’re right there with you. *Remote testing* has many advantages that cannot be ignored, such as the following:

- Testing can be performed nationally/internationally at a low cost.
- Travel times are eliminated.
- You can save on expensive equipment (e.g., laboratories).

The advantages are self-explanatory; for many companies and startups, these criteria are decisive for remote testing. However, every light casts a shadow. High flexibility goes hand in hand with some disadvantages, such as the following:

- Remote testing depends on the technical infrastructure of the users.
- With remote testing, you risk losing important information if, for example, no webcam is available, which may be the case depending on the target group.
- You have little or no control over whether users honor confidentiality agreements.
- You cannot test non-digital products without some distortion, although distortions are also possible with digital products.

Since these disadvantages are less known and less obvious, let’s further discuss them to give you a better basis for your decision. After that, we’ll have a look at what role the type of product plays.

12.4.1 Dependence on Technical Infrastructure

Especially if you’re developing a technical product, you and your team probably have at least solid technical equipment. You probably have reliable Internet, sufficiently fast computers, and access to other technology you need to develop your solution.

This criterion is not always the case with your target groups: If the necessary infrastructure is missing or cannot be operated as well as required for testing, remote testing becomes difficult, if not impossible.

Attention, Fallacy!

Whenever something is self-evident to you, you tend to assume that this fact also applies to others. However, making inferences about user equipment based on your own equipment can also be plagued by fallacies.

In our early days of remote testing, we were sometimes completely amazed when participants did not have a stable or only a very slow Internet connection or did not have a webcam. Sometimes, the problem was that participants were not allowed to install anything on their PCs because it was not their own, and they lacked administrator rights. Commonly, users could not operate their own technology as we would need them to in order to carry out the test.

12.4.2 Loss of Important Information

Imagine performing a test where you observe usage and interview the person. Unfortunately, their Internet connection is slow and often hangs, and your counterpart doesn't have a webcam either. As a result, you lose a lot of information. Maybe you don't even see whether the task was successfully solved. At the very least, however, you're missing non-verbal information conveyed in regular interviews via body posture, facial expressions, and pauses. The less real data you have, the more you have to guess what is happening. And the more you guess, the less you get out of the test.

In any case, make sure you have the right basic technical equipment! If you work with a target group living in a rural area or with little technical equipment, testing a few people with the right equipment is useless. If you follow this approach, you'll most likely get *biased* data. Your participants will likely have different technical requirements and knowledge than the target group; thus, the results would not be transferable or would only be partially transferable. This problem is not so easy to solve.

12.4.3 Less Control over Confidentiality

The third major disadvantage is that you have less control over what happens on the other side. If you invite test participants to your labs or rooms and give them a device with the prototype preinstalled, then you'll see what happens to it.

If, on the other hand, you let participants test remotely and share the screen or provide a product in advance, then you have little influence on what they do with the prototype. You can neither see nor prevent the screen from being recorded or photos being taken. However, great value is often placed on confidentiality, especially in early phases or with innovations. Confidentiality is much more difficult to guarantee when you work remotely than when testing in person.

12.4.4 Non-Digital Products Cannot Be Tested without Bias

The last major drawback is that remote testing products that are at least partially hardware is hard to do without skewing the results. (We assume you do not want to distribute the products freely yet.)

For example, if you're developing a smartwatch or a smart refrigerator, you can check the user interface (UI) for *comprehensibility*. However, identifying problems that arise through *interaction* with the smartwatch or the refrigerator itself will be tough. For this task, a user must wear the smartwatch on their arm or stand directly next to the refrigerator. Thus, your test results will likely be too positive for new development and will not consider the whole interaction area. Remote testing of (even partially) physical products is thus a bit like turning a blind eye to testing.

A Personal Example: What Remote Tests Miss

We personally send and receive a relatively large number of parcels and packages. In the past, receiving these parcels was a big challenge because, due to our work, we were often not at home when the delivery was made. Now, parcel lockers exist to make our lives easier in many dimensions.

But if you were to watch Michaela operate a parcel locker in Germany, you probably wouldn't believe it: Somehow, the touchscreens don't work for her. The buttons don't respond, and she often presses so hard that her fingers bend. Sometimes, she presses the buttons with four fingers at once until they respond. This problem would never have arisen via a remote test (and obviously wasn't figured out) because the problem does not reside in the logic of the digital UI but in the interaction with the physical part of the parcel locker product (the touchscreen).

Thus, remote testing only gives you part of the truth when you have a product with a physical component.

Don't assume this list of disadvantages makes the case against remote testing—we favor and use remote testing regularly. The only questions are when and for what. Because this choice is so important, we would like to provide a few guiding questions to help you decide when to conduct a test remotely or in person. When deciding for or against remote testing, some questions to ask yourself include the following:

- **What kind of product is it?**
  The first—and probably most crucial—question is what kind of product your solution is. If you have a purely digital product, like an app or a web application, the signs for remote testing are good for now. Assuming your target audience has the right equipment. Accordingly, the second question we recommend is, "Does the majority

of your target audience have the right equipment to participate in remote testing?" If they don't, we recommend either testing in person or considering how you can easily provide the target audience with the right equipment. You could also decide to forgo the matching equipment and lose some of the information if necessary.

- **What information do I need to decide on the quality of my product?**
  If you have a physical product, you should ask yourself what stage of development the project is currently in. Remote testing can be useful if you're still at the very beginning and the goal is coming up with a rough concept or understanding the problem. Remote testing is less appropriate if you already have initial physical prototypes and are concerned about haptics and handling.
- **How important is confidentiality in my test?**
  The final thing you should ask yourself is how important confidentiality is in your case. The more secret the project is, the more likely you should test under conditions that you can shape yourself. In most cases, this question will make the decision in favor of testing in person on your premises.

12.4.5 Conclusion

A final word at this point: Many reasons speak in favor of remote testing, and many reasons speak in favor of joint on-site testing. You must weigh which reasons are more important for yourself and your company.

In general, however, some food for thought is that on-site interviews are often easier to conduct and often produce better results. Interviewers can observe more of their interviewees (posture, facial expressions, etc.) and are thus better able to adjust to them. Especially in solutions where a lot of trust is involved, this difference can be decisive for the quality of the results.

12.5 Designing Test Cases

Before we turn to the individual methods, we would like to highlight a topic that belongs to some of the methods we'll present in Section 12.6. We'll deal with it appropriately in this section and then, if necessary, take it up again in the individual methods. The topic in question is "designing test cases." In this context, we mean the design of individual tasks used in a user test (e.g., a usability test or a usability sprint) and the specification of what is to be tested by the user.

We follow a systematic approach for our user tests in which we define the test cases the same way for all users in a group. In our user tests, all people are given the same tasks in the same setting. This premise is essential to collect objective data on whether a design solution is well understood and can be successfully operated by the user. By making the test cases uniform, we ensure that the observed errors can be systematically analyzed and improved. To ensure this uniformity is as simple as possible, we describe the test cases using several dimensions:

1. Preconditions (input state)
2. Instructions for the user
3. Correct execution of the task
4. Postcondition after correct execution (end state)

12.5.1 Preconditions

The *preconditions* describe the state of the prototype when the user comes into contact with it. In this context, we define, for example, which page or screen in a prototype is started or whether and which entries have already been made on the screen. This procedure ensures that all users in the test have the same initial condition. If this were not the case, we would be unable to make clean comparisons later. We would not know, for example, whether an error occurred due to alternate start conditions or due to user interaction.

The preconditions (i.e., the *input state*) are not explicitly communicated to the user but are only crucial for the test manager to create the right framework conditions. When you plan your test cases, an important part of the preconditions is also to specify the appropriate usage environment. We always try to be as close as possible to reality—at least with the most important framework conditions.

An Extreme Example

Imagine you're developing a system for an intensive care unit. Then, the ambient volume, the displays from various other systems, the alarms from various other devices, and the time pressures are relevant—but not the question of whether the wallpaper is purple or blue.

We try to design the critical elements into the scenarios and the environment—on the one hand, by equipping the test rooms with the appropriate equipment and, on the other hand, by including disruptions, time pressure, and the like when necessary. We might also include, for example, working in the correct color space during the design to display the color contrasts correctly on the test and later equipment. (Your computer, on which the UI is created, and the prototype may have a better resolution or contrast ratio than the later product.)

Conclusion

The systematic and correct design of the preconditions is important to obtain results that allow statements about the quality of the design solutions. We hope this section has given you some insight into the various points to consider when planning tests. Only you can decide which points are relevant in your specific case based on your situation and objectives, as these choices are closely interwoven with the system, product, or service in question and the usage environment.

If unsure, you can look at the introductory section on the context of use (see Chapter 2, Section 2.2.5) and Chapter 9 on the context of use analysis and check which aspects could influence the use of your system, product, or service. These aspects are precisely the aspects you should consider as preconditions.

12.5.2 Instructions for the User

By "instructions for the users," we mean everything a user is told or shown. This concept usually includes a description of the initial situation and a description of the task (i.e., a scenario they are supposed to imagine for the test).

An example of such a scenario could be the following:

> *"Please imagine that you have been having eye problems for several months. Your eyes often feel tired and watery. Now you're looking for a suitable specialist to look at your eyes and help you take appropriate steps. As you're googling for it, you come across the following ad. Please read the ad and tell me in your own words what you're being shown here."*

We always try to formulate the scenarios so that it is easy for the user to think their way into the respective situation and the task is clear. Nevertheless, the description should be as neutral as possible and not contain any influence in one direction, otherwise, you'll directly distort the result.

An example of a less successful scenario would be the following text:

> *"Please imagine that you have been having eye problems for several months. Your eyes often feel tired and watery. Now you're looking for a suitable specialist to look at your eyes and help you take appropriate steps. A few weeks ago, a friend of yours discovered an app that helps find specialists. He is completely delighted with the app and highly recommends it to you. Please download the app 'Healthyfy' once, and install it."*

The combination of "friend" and "recommendation" already creates a positive influence, which you should avoid in a test situation.

You should additionally formulate scenarios so that almost every person in the user group can put themselves in the scenario. Thus, the wording must not contain too many details, for example, from one's private life. Think of a horoscope: Somehow, you should feel that it *could be* an actual situation from your life, but it isn't.

To make the scenarios as realistic as possible—and thus the results as good as possible—we recommend you take your time describing the scenarios in the test cases and build them up as logically as possible. Users should, therefore, first be asked to download the app and log on before they are asked to solve tasks in the app. The logic of "from the coarse to fine" also applies in this context. Start in general and only go into more detail step by step.

Skip Steps?

Of course, you don't have to have all tasks performed every time in your tests. But you should be aware that you'll have to explain the skipping of important and necessary tasks ("Imagine you have already downloaded the app and created an account. You now want to start the app for the first time.") and that users may sometimes lose important information as a result of this skipping, leading to problems later on.

For example, let's say you have a tutorial that explains the product you want to test. You have already optimized the tutorial, and you know that it works. Nevertheless, skipping the tutorial in the test doesn't make sense because you would miss knowledge about how to use the product in the following test.

One last tip: Don't give instructions to users by speaking freely. Although ad hoc may seem more elegant, you'll notice (should you ever do it) that, with eight people, you'll use eight slightly different terms and tell eight slightly different scenarios. Now, again, you have the problem of starting conditions that are different. This discrepancy might not have any effect—but what if it does? Therefore, we recommend reading the scenario aloud. Then, several interviewers can be tested in parallel more consistently.

12.5.3 Correct Execution of the Task

We describe the correct execution of the task to make it easier for the test administrator to decide whether the task was executed in the best possible way or whether the users took a detour. This description also makes it easier to ensure that all tests (even with multiple interviewers) are run the same way.

The steps for the correct execution of the task are not visible to the users. They are an internal aid that help test administrators control the follow-up questioning on the individual tasks in a target-oriented manner. In addition, documentation during and after execution is more straightforward if it is clear when a task is assessed as "completed," for example. We'll come back to this idea in a moment when we discuss post-conditions. There, however, we'll deal with the end state (i.e., the end of the task). However, errors that are no longer recognizable afterward can also occur during the solution.

For example, usability tests must also be carried out for the approval of COVID rapid tests, and we have carried out several thousand of these tests. One important step is to document whether users have extracted the biomaterial correctly. If, for example, the swab was placed in the mouth instead of the nose (in a test that is not intended for this purpose) and the test was otherwise performed completely correctly, the entire test is still invalid.

12.5.4 Postcondition after Correct Execution

The situation is similar to the *postconditions*—that is, the end *state* when the user has completed the task. Describing the output state helps to compare whether there are any errors quickly and, if so, what they are.

Especially for fast methods, where a first result should be available shortly after the test execution (for example, because the next revision and test phase is firmly planned), this approach is helpful because the assessment of whether a test case has been solved successfully or not is quick and easy.

12.5.5 Determine the Number of Test Cases

How many test cases you can cover in a test, and the ratio of actual testing to interviewing, depend entirely on your product and the stage in its development:

- The longer and more elaborate the scenarios are, the fewer scenarios you can test meaningfully.
- The closer you are to releasing the system, product, or service, the more you should focus on objective data and the less time you should invest in surveys—provided you have asked the right questions beforehand.

12.6 The Methods Presented

Before confronting you with each method, we'll provide a brief overview of the methods you can expect and how each method section is structured. You'll encounter the following methods:

- Usability and UX testing (Section 12.7)
- Usability and UX sprint (Section 12.8)
- Rapid user test (Section 12.9)
- A/B testing (Section 12.10)
- UX benchmarking (Section 12.11)
- 5-second test (Section 12.12)
- Standardized questionnaires (Section 12.13)
- Analysis of quantitative usage data (Section 12.14)
- Web tracking/click analysis (Section 12.15)
- Diary study (Section 12.16)
- Card sorting (Section 12.17)
- Expert review (Section 12.18)
- Association space test (ARTTest) (Section 12.19)

- Eye tracking (Section 12.20)
- Physiological measurements (Section 12.21)

These methods are a mixture of well known and widely used methods (e.g., the *usability and UX test*, *card sorting*, or *eye tracking*) as well as our adaptations of various methods that have supported us well in our work over the past years (*usability and UX sprint* and *ARTTest*).

To help you orient yourself among these methods, we have agreed on a fixed sequence for the presentation of the methods, which remains the same for (almost) all methods:

- At the beginning of each method section, we'll briefly introduce the method in general.
- This brief introduction is followed by an explanation of the steps that should be implemented using this method ("How Does It Work?").
- We then explain under which conditions it makes sense to use the presented method ("When Is It Used?").
- Next, we provide an overview of the most important things to perform the method ("What Do You Need?").
- Advantages and disadvantages are presented in detail ("What Are the Advantages?" and "What Are the Disadvantages?").
- The last section in each case shows which methods would have similar or partial results ("What Are the Alternatives?").

Each method section closes with a profile of the method. To see all the methods at a glance, we recommend looking at our method matrix in Chapter 14, which provides an overview of the suitability of the individual methods.

Following the presentation of the methods, we'll also look at how you can analyze errors and derive reasoned decisions for the further development of your project based on your results (Section 12.22 and Section 12.23). This chapter ends with an extra section on medical device evaluations and examples for moveHUB, BusinessBooster, and Healthyfy.

But for now, let's get started on the first method!

12.7 The Usability and UX Test

The usability test is the first and probably most well-known method we want to introduce in this chapter. Our initial focus lies on *usability*—the UX will be added later in this section.

Usability testing is a combination methodology consisting of observation and interview techniques (Sarodnick & Brau, 2006). In its purest form, however, pure observation is also possible. Observation in the context of a usability and UX test is not free but is oriented toward tasks the users receive from the test leader.

12.7.1 How Does It Work?

The way usability tests work can be divided into several parts, which we'll discuss in this section.

Before the Test

In a classic usability test, a person from the user group of the system, product, or service is confronted with the application. The basic prerequisite for a successful usability test is the presence of the right user group.

To ensure we get the right user group, in the preparation of usability tests, we always work with the development of screeners. A *screener* is a document full of open-ended questions that enable the person asking the questions to decide whether or not the respondent belongs to the user group (see also Chapter 9, Section 9.3.3). We used the screener already during the pre-selection of the participants and not only during the execution of the usability tests themselves. In this way, we can ensure that we avoid wasting time on unsuitable people participating.

For some applications, *more than one relevant user group exists*. In this case, we recommend creating a separate screener for each user group so that you do not end up starting the study with a member of a mixed user group. Still, we can determine which group a participating person belongs to. For each user group with different tasks or different characteristics, we recommend inviting at least six participants to conduct a usability test. Even though this sounds like a relatively small number of participants, studies have now proven that even with this number of participants, a large percentage of errors can be found, as shown in Figure 12.4.

As shown in this graph, the probability of discovering a usability problem drops significantly with fewer people and even more with fewer than five people. With more people, on the other hand, the probability of discovering a usability problem increases significantly (for example, with six users already, at least 60%, over 85% of the errors are found on average). Of course, at some point, the costs and benefits will have to be weighed against each other.

If alternatives need to be compared, the participation of significantly more people helps observe statistically relevant differences in the evaluation. Usually, you would work with at least 30 participants if you want to recognize statistical differences.

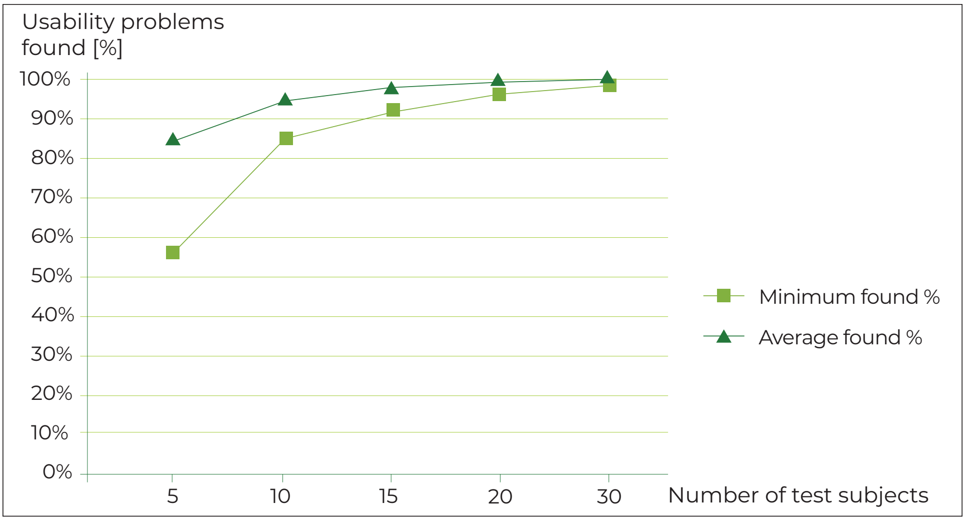


Figure 12.4 Percentage of usability problems found over the number of test subjects (based on data from Faulkner, 2003; graph based on FDA Guidelines on Applying Human Factors and Usability Engineering to Medical Devices).

The number 30 comes from the fact that statisticians assume that various forms of distribution approach a normal distribution at 30 persons. This assumption does not mean the answer to all questions is "30." If you want to know how many people should participate in your test for a clear distinction, we recommend using *sample size calculators*. You can find a sample size calculator at *https://www.surveymonkey.com/mp/sample-size-calculator/*.

However, these considerations are mostly only relevant for evaluating questionnaire data. For the evaluation of objective data from observation and the detection of errors, a smaller number of participants is often sufficient.

After completion of the screener and recruitment of the participants, we create a *test guide*. The test guide aims to make the results comparable for all participants and contains all tasks, including the following:

- A description of the preconditions ("Is the application already installed? Is the user already logged on?")
- The task guide for the user
- The postcondition ("The app is installed on the smartphone, and the user has logged on with their credentials")

As described in Section 12.5, the task instructions for the user are primarily defined as a scenario to get closer to a real situation. The initial situation is briefly described. The scenario description ends with a request to perform a specific task and let the test operator know when it has been completed. An example scenario might be the following:

> *"Please imagine that you received a call from your bank yesterday that all customers must use the bank's online banking in the future. The bank employee informed you that you'll receive another letter with instructions on how to do this. This letter has reached you today, and you have decided to enroll directly. Please read the letter and then set up your online banking access. During the task, please speak anything on your mind, and let me know when you have successfully set up your online banking."*

In addition, the test guide contains all the questions to be asked of the users and all the instructions to test operators so that they know which material is to be used, for which condition, and when.

Let's look at a small excerpt from a sample test guide as we would use it. The text in the square brackets contains the instructions for the test operator; the text in the curly brackets asks the test operator to adopt the specific wording for the participant for this case.

Initial Situation

What are you currently doing when {core task to be solved by product}?

- Do you always do it the same way?
- [If no:] In what situations do you do it differently?
  - How do you do it, then?
  - [Repeat block of questions until participant stops listing other situations/types.]

During the Test

The test can begin once recruitment has been completed and the guide has been written. Depending on the complexity, the test series sometimes starts with a pre-test, where we invite one person a day or a few hours before the other participants to see if our test guide will work out.

As soon as the participants arrive (in person or remotely), the test starts with the redelivery of the privacy policy, the confidentiality agreement, and the consent statements of the participants to having the test recorded. You don't have to do record video, but the video can help you greatly when reviewing specific sections in detail afterward. Now, we're ready to start with the content.

At the beginning of the study, we first ask (again) for the *screener criteria*. This step enables us to notice short-term changes and exclude people if necessary. The actual *test* then starts *according to the interview guideline*, in which the users are presented with the scenarios and are told to solve the tasks without assistance from the test leader. While they perform the tasks with the application, the users are observed—either directly by a moderator or indirectly via a camera, as shown in Figure 12.5.

In addition, users are usually asked to "think out loud" during use and say what is on their mind. We ask that the thoughts are not filtered beforehand but spoken out freely.

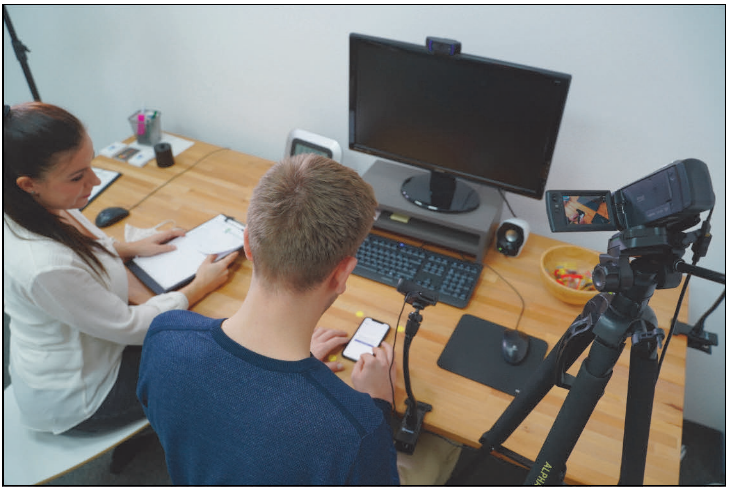


Figure 12.5 A typical usability test with a simple experimental setup.

In addition to task execution and thinking aloud, usability tests are often combined with follow-up surveys. In the follow-up surveys, further questions are asked. You can use freeform questions that again include a subjective assessment of the ease of use and appeal of the application, or questions based on a scale can also be used. If time permits, we also ask whether the participants felt they had successfully completed the individual tasks and inquire about observed errors and their suspected causes.

A usability test can be carried out with finished products and prototypes. The only important thing is that the tasks can be implemented with the prototypes provided. To ensure that the tests significantly impact product development, product managers should observe the tests live.

After the Test

After the execution, we've come to the evaluation of the collected data. In the evaluation step, different measures can be considered, such as the following:

- Number of solved or unsolved tasks
- Number of errors made
- The severity of the errors made
- Observed difficulties in implementation that did not result in errors or unsolved tasks
- Subjective evaluation of the application
- Subjective evaluation of the success of the task
- Subjective justification for the errors

Theoretically, you can record many more measurements, for example, the times required to perform the individual tasks. In practice, however, we have found that the *actual operating time* is only relevant in rare cases.

We only consider time when we have an objective reason for doing so. Reasons to consider time, for example, include when the gaze aversion time from the street had to be taken into account or the turnaround time of passengers undergoing airport security checks to avoid long lines. In most other cases, however, we prefer to look at "perceived time" or "perceived ease" from the user's point of view because this concept has proven to be a much better measure for subsequent evaluation than the actual measured time.

Extension: UX Test

Up to this point, we have described a classic usability test. This type of usability testing is valuable and should be used at various points in development to ensure that the operation of the application works smoothly. However, we have often found that focusing on usability is not enough, especially if your development is the initial developmental stage of an application.

In these cases, we regularly extend the usability test into a *usability and UX test*. Thus, not only do we test whether a task can be executed well and successfully, but we also pay attention to whether the application solves the right tasks and whether the application solves the user's problem. We achieve this test by adding additional parts to the usability test:

- The first supplement takes place at the very beginning of the test and asks about the problem and how it is currently being dealt with. This information can be gathered either in the form of a semi-structured interview or, for example, with the help of a comic strip in which we present the problem and ask whether the initial situation is known and how it is currently being dealt with.
- The second addition comes after the usability test and asks whether the presented application solves the problem or at least contributes to the solution. In this way, a test provides us information about *usability* and the *problem-solution fit* from the user's point of view. This information us usually gathered the form of a semi-structured interview, sometimes combined with scale questions, for instance, "On a scale of 1 (not at all) to 5 (very much), how much does the presented application contribute to solving your problem?".

In any case, we recommend that you pay close attention to the evaluation of usability and UX tests and also make it clear in the documentation whether results are based on observations (and thus is objective data) or based on surveys and statements by participants (and thus is subjective data).

Unfortunately, it often becomes apparent in this context that the two data types do not have much to do with each other. Thus, participants often give fantastic subjective ratings but cannot solve the task objectively. This discrepancy may not seem critical at first. However, it means you'll later receive extremely negative feedback or must provide a high support effort.

Conversely, great usability feedback and poor subjective ratings will result in your application being neither purchased nor recommended. Therefore, to understand what you need to work on, you must be able to separate both data types flawlessly. This separation also helps with prioritization.

12.7.2 When Is It Used?

Usability and UX testing can be used at different points in the development of products. The earliest possible time to conduct a usability and UX test is the completion of initial wireframes or prototypes that show the product, system, or service from the user's perspective.

These wireframes and prototypes can be quite simple and do not have to meet any special requirements, except that what is shown is later what the users would see and use. Since the goal of a usability and UX test is to find out how well the use works from the user's point of view, the representation must also be done from the user's point of view. Schematic drawings of the system architecture are, therefore, not sufficient. But as soon as your first wireframes exist, you can start testing.

If you only have wireframes and no prototypes, then the test works from the basic structure precisely as described. The only difference is that, instead of clicking somewhere, participants describe what they would do, and the test leader then executes this action by switching to the appropriate wireframe. This procedure works even with paper prototypes.

The earlier you use usability and UX testing in development, the sooner you get feedback on whether the development of your application is going in the right direction. This rule is all the more true when considering UX, problem-solution fit, and usability.

In addition to conducting usability and UX tests in very early phases of development, they can also be used as a final evaluation at the end of development. In this case, usually marketing, not the development department, drives this round of tests (Section 12.1). The focus is then even more intensely on showing what added value the application brings from the user's point of view and how well this application solves the user's problems. In some companies, however, usability and UX tests also serve as quality gates in development after development is complete, at which point the development team must prove (by means of a neutral test) that the quality of the application meets the company's standards.

Finally, usability and UX tests can also be wholly detached from development, exclusively to check the current state of a solution on the market. In this case, the term *UX benchmarking test* is more commonly used, but we'll cover this topic later in Section 12.11 on its own due to its importance for the development decision.

12.7.3 What Do You Need?

You don't need much to conduct usability and UX testing. In the classic setup, a usability lab can be viewed via one-way mirrors or cameras from an observation room, as shown in Figure 12.6.

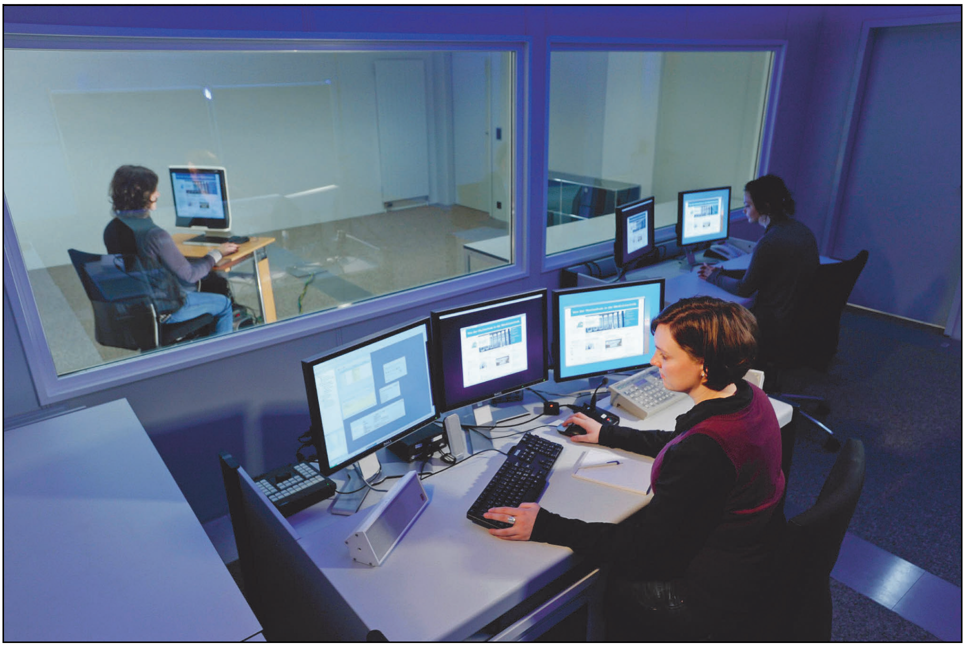


Figure 12.6 Classical usability lab with one-way mirror, at the Zurich University of Applied Sciences. (Source: https://www.zhaw.ch/; credits: ZHAW Applied Linguistics/Frank Brüderli)

However, one-way mirrors have now mainly been replaced by digital methods such as broadcasts via camera. With the current infrastructure, you can completely separate the laboratory and observation areas (with one-way mirrors, the areas must be next to each other). This setup has an advantage in that observers can follow the tests from anywhere in the world without being cooped up all day in a dark room with no light so that the users do not realize they are being observed.

No matter which solution you choose, the minimum lab requirement is to provide the application in a form that is as close as possible to its eventual form and a device to observe usage. So, in theory, you could provide a link to a prototype that users then open on their smartphones as long as you can observe usage.

From our point of view, you need, as a minimum for a usability and UX test:

- A list of criteria for selecting suitable participants for your test (screener)
- A test guideline that specifies which tasks the users have to complete and which questions you want to ask as a minimum

- An initial representation of the application from the user's point of view (wireframes are sufficient)
- A possibility to observe users during use
- Documentation of the results

There are always different "extension levels" for the material mentioned. The screener can be just a sticky note with the age range noted if age is the only important criterion. Or you might have a six-page document with 17 questions and associated answer categories. The test guide can be just a list of two items and three questions, but it can also be a 20-page document in which everything is worked out to the smallest detail. The same advice applies to the prototype, the observation opportunity, and the documentation. Adjust the scope to fit your needs.

In our experience, the need for this test is strongly related to whether you'll have to justify yourself to anyone afterward regarding the implementation, the results, and further steps in the development. The sooner this need arises, the more effort you should invest in preparing and documenting all steps. Typical conditions under which justifications are always necessary include having a quality management system or operating in a regulated industry.

12.7.4 What Are the Advantages?

Usability and UX tests are particularly powerful because they give you insight into subsequent user behavior even before you have put any effort into technical development. In this way, you can determine rather early whether your application fits the user group's requirements and how well it has been implemented. You can also compare different implementations with little effort and choose the best one.

Another advantage of usability and UX testing is that you gain objective data by observing usage, whereas many other market research methods only generate subjective data. Thus, you are not (only) gathering opinions that the product is easy to use, but you can clearly show, with a usability and UX test, that the product is faster, easier, and better. Combining observation and survey enables you to make targeted revisions to the design to help reduce the observed errors, which is usually not possible with pure surveys.

12.7.5 What Are the Disadvantages?

One disadvantage of usability and UX tests is that they can only be meaningfully carried out once initial prototypes are already available from the user's point of view. Depending on the project, a relatively large amount of time may have already been spent developing and coordinating the underlying concept. Admittedly, this is more a project setup problem than the usability and UX tests themselves. Nevertheless, we would like to mention it because it happens quite often in practice.

Another disadvantage is that the simulated usage environments can sometimes produce different results than if the same task were observed in reality. Social factors, in particular, such as the presence or absence of other people during use, can significantly influence test results. However, this disadvantage is negligible compared to the need to wait until completion.

Compared to questionnaires and purely online surveys, implementing and evaluating usability and UX tests are relatively time consuming since a test is usually conducted with one participant and one test leader. Especially in the case of products with different target groups or complex products, you might end up with a large number of participants or a lengthy implementation period relatively quickly. In addition, an evaluation of the observed and surveyed data is carried out afterward. For this purpose, it may even be necessary to watch video recordings of the tests again—depending on the desired level of detail for the evaluation.

12.7.6 What Are the Alternatives?

Various alternatives exist to using usability and UX tests depending on the objective. Suppose the focus is to remain on observation, for example. In that case, usability and UX sprints (Section 12.8), A/B testing (Section 12.10), rapid user tests (Section 12.9), or UX benchmarking (Section 12.11) can be used. Standardized questionnaires (Section 12.13) or focus groups (see Chapter 9, Section 9.3.4) can be used to understand user opinions.

Table 12.1 shows the profile for this method.

Profile: Usability and UX Test	
Brief description	Usability and UX testing is a method in which users are confronted with a (prototypical) solution and are asked to solve specific tasks with it, usually while thinking aloud. Afterward, they are asked about their experiences with the solution.
Advantages	■ Enables usability testing before solution completion. ■ Enables objective data to be obtained on problems in use and subjective data to be obtained to evaluate interaction. ■ A supplementary survey strongly supports the subsequent optimization of the solution.
Disadvantages	■ Requires a prototype that replicates the interaction for users. ■ The simulated environment influences the results—the worse the simulation, the stronger the influence. ■ The implementation and evaluation are relatively complex because only one person can be observed and interviewed at a time.

Table 12.1 Profile for the usability and UX test method.

12.8 Usability and UX Sprint

The usability and UX sprint is a smaller version of the usability and UX test and is a different evolution for the agile application.

We usually adopt usability and UX sprints when the execution of various iterations is planned directly, and the development cycle is short. In most cases, we shorten the test execution and evaluation to one day to obtain results that can be used in further development as quickly as possible.

12.8.1 How Does It Work?

In terms of its basic structure, a usability and UX sprint follows the same procedure as a usability and UX test. The differences are mainly found in the preparation, the number of tests performed, the evaluation of the tests, and the documentation.

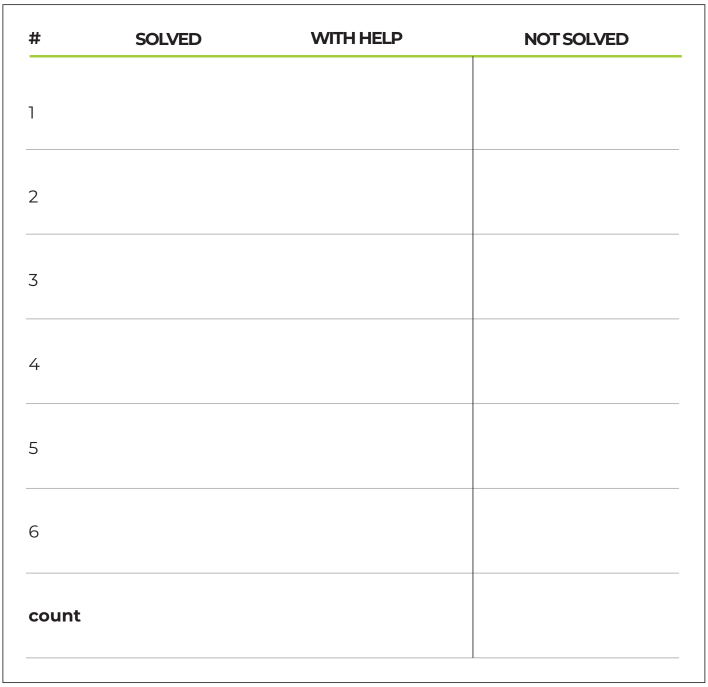
For usability and UX sprints, a fixed *rhythm* is usually established in which the tests are performed. For example, in the case of two-week sprint cycles with the start of a new sprint on Monday, the test execution and evaluation for the usability and UX sprints are always scheduled for every second Thursday so that the results are available in time for sprint planning on Friday.

As soon as the rhythm has been clarified and the rough content for the sprints has been planned, people are invited to participate directly in the upcoming sprints. This schedule ensures that participants are available in good time, even for very short sprint cycles. This *pre-invitation* becomes even more important the more complex a target group needs to be found for usability and UX sprints. If you plan an application everyone can use, you can most likely start your search two days in advance. However, if you need internal or external specialists, then they may not have time to participate in your testing within less than two weeks.

Since the execution and evaluation of the usability and UX sprints have to happen within one day, only a few people can participate in the tests due to the time alone. In most cases, we reduce the execution time to about 45 minutes (compared to 60 to 120 minutes in a regular usability and UX test). Nevertheless, no more than six people can usually participate on one day since short breaks are needed between the individual tests, and a joint evaluation will occur afterward.

Because the results are needed the next day, usability and UX sprints always occur with the development team. The people who work with the results must therefore be present as observers. This approach also makes the parallel execution of several tests less useful because, otherwise, some of the information for the later weighting and evaluation of results are only indirectly available.

The *evaluation* of the usability and UX sprints usually takes place live. For this purpose, we have developed templates on which the task is displayed, and a line is shown for each participant, as well as a column for “Solved” or “Solved with help” and “Not solved,” as shown in Figure 12.7.



#	SOLVED	WITH HELP	NOT SOLVED
1			
2			
3			
4			
5			
6			
count			

Figure 12.7 Sample document documenting the main results of a usability and UX sprint.

Each observer receives one of these slips of paper and can then write down, live during the test, which participant did which task and what worked well or did not.

After each test execution, the results are summarized on a central sheet so that all the critical information for the *use case* is collated, and you can see at a glance how often it was solved or not solved. This knowledge is used to set priorities in the subsequent workshop, as shown in Figure 12.8, where you’ll decide where work should be done and find revision options for the “worst” implementations directly with the development team, which flow into the next sprint.

Since everything in a sprint is designed for speed, we usually keep the time for documenting the results relatively short and present the core results for the individual use cases on one page each, supplemented by the solutions found. The result is more of a working document intended to record the current status and define the next steps than documentation in the classic sense.

Of course, you can set other priorities for the evaluation and documentation. A usability and UX sprint is not a fixed method, but a variant we have successfully used in recent years to meet the ever-faster development cycles. In this approach, you can adjust the interval between tests according to your needs. We’ve had everything from four weeks at a relatively leisurely development pace (is that still a sprint then?) to daily sprints that led directly into rework, but then only on a prototype basis.

Figure 12.8 Working out revisions and design solutions in the joint workshop.

12.8.2 When Is It Used?

We primarily use usability and UX sprints as a usability and UX partner with an agile development team. However, this method would work just as well if an internal usability and UX team worked with an internal agile development team.

By planning development work in sprints, it is often necessary to get feedback on existing concepts and designs within a very short period to avoid developing in the wrong direction for an unnecessarily long time. Our usability and UX sprints then usually take place at the end of a development sprint so that the results from the sprint are available for the sprint planning of the development.

12.8.3 What Do You Need?

The requirements for a usability and UX sprint are the same as for a usability and UX test—with one exception: You also need the presence of the development team. Since the results are only recorded live, all participants must directly see what happens in the tests.

This presence is not necessarily required with conventional usability and UX tests, which have extensive documentation, quotes, and, in some cases, video excerpts of the tests. Thus, for a usability and UX sprint, you'll need the following:

- A list of criteria for selecting the appropriate participants for your test (screener)
- A test guideline that specifies which tasks the users have to complete and which questions you want to ask as a minimum

- An initial representation of the application from the user's point of view (wireframes are sufficient)
- A way to observe the users during use
- The development team, which is live during the implementation and evaluation
- Brief documentation of the results and next steps

12.8.4 What Are the Advantages?

The biggest advantage of a usability and UX sprint is *speed*. By compressing all execution and evaluation activities into a single day, results are simply available for further work much sooner than is the case with other formats. Another positive effect of usability and UX sprints is the closer *involvement of the development teams*, as they are usually on-site—again, due to the high time density. As a result, all team members can better understand decisions and usually stand united behind the decisions. This is less often the case with other formats, where the results are significantly less direct.

If usability and UX sprints are integrated into agile development, then the influence of the test results on the implementation is significantly greater than with detached test activities. This is especially true if the usability and UX sprints occur regularly during the development sprint. In the past, we have often seen a reduction in the need for internal discussions because critical discussions are no longer held between members of the development team; instead, due to the high availability of regular tests, the discussion points are simply introduced as test content.

12.8.5 What Are the Disadvantages?

Usability and UX sprints are significantly shorter than traditional usability and UX testing, and fewer people participate, which can make getting sufficient feedback from all target groups difficult within a sprint. As a result, you risk uncovering fewer usability issues than would have been the case with a regular test with more participants. In most cases, this limitation is not critical because the individual usability and UX sprints are more frequent and thus provide multiple opportunities for testing. Thus, if you're testing comparable content, you can increase the number of tests over the iterations and thus get similarly good results.

However, as a result, you may need more time to obtain feedback from a sufficient number of participants from each user group. If you don't have the luxury of extra time, running several tests in parallel is a good idea. However, this approach naturally loses the advantage that all participants have seen all tests.

Another disadvantage of usability and UX sprints is the greatly reduced documentation. Particularly for internal justification, generating sufficient documentation while being fast enough is difficult. A possible solution is to record usability and UX sprints on video and start the detailed evaluation and detailed documentation afterward so that they are not available for sprint planning but are available later.

12.8.6 What Are the Alternatives?

If speed is not the most important thing, then instead of a usability and UX sprint, an ordinary usability and UX test (Section 12.7) can be used. If speed is essential and the criteria are not very demanding on the target group, then rapid user tests (Section 12.9) can be used. If only rapid feedback on a preferred design is required, but not necessarily on the operation itself, then online surveys with standardized questionnaires (Section 12.13) can be used.

Table 12.2 shows the profile for this method.

Profile: Usability and UX Sprint	
Brief description	The usability and UX sprint is a shortened version of usability and UX testing, designed to be used regularly in agile solution development, enabling more tests to be performed over time. The execution and evaluation usually happen within one day.
Advantages	▪ Higher speed compared to the classic variant, with more participants. ▪ Regular monitoring of agile development projects, as integrated into the sprint cycles. ▪ Less need for internal discussion because sprints regularly enable the objective clarification of open questions.
Disadvantages	▪ Usually shorter than classic usability and UX tests with a thus limited test focus. ▪ Fewer participants per test, which means that the most critical interaction elements should be checked in several tests. ▪ Less detailed documentation, as more of a working document and thus less suitable for internal reports.

Table 12.2 Profile for the usability and UX sprint method.

12.9 Rapid User Tests

A rapid user test is a slimmed-down version of a usability and UX test carried out remotely via a platform. A test can be prepared in advance via a digital platform and then presented to the users automatically or, if desired, by a moderator. Due to the high degree of automation, the test results can be available within a day.

We have included rapid user testing in this chapter (even though the method is not described in the literature) because it is becoming increasingly important in practice due to the speed with which tests can be performed. At the current time, only a few vendors provide this type of testing. However, we assume that similar testing methods will become more critical in the future due to the increasing level of digitization.

12.9.1 How Does It Work?

Rapid user tests are designed to test web-based content quickly. For this purpose, a test concept can be created independently on a platform that starts at a specific URL. Afterward, you have the opportunity to enter the tasks that would otherwise be given to the participants by a moderator. Then, you can define individual questions for the post-survey, which will be displayed in the follow-up to the tasks for the participants.

Selecting the desired target group is relatively simple, based primarily on age and gender. Other criteria can be added, although the speed of the tests can drop significantly because these criteria can no longer be queried automatically.

If everything is set up as desired, the test is booked, released online, and carried out automatically. Test execution is recorded with the help of *screen recording*. In addition, what the participants think out loud during the test is recorded so that an explanation of the behavior is also given.

The result is the video recordings of the tests, which were created automatically. The results are usually available within a day, so you only need to review and evaluate the videos yourself. To make the best use of the results of a rapid user test, you should plan the time and expertise that will be needed for the evaluation after the test has been carried out. In this area, we repeatedly observe that companies have difficulties translating customer problems into successful design revisions.

Thus, if you're running a rapid user test, don't assume that the test will be completed in a matter of hours. While you'll have the results in videos, you must invest enough time with your team afterward to understand what worked and what didn't and how you want to handle it. This evaluation doesn't necessarily have to take a long time; depending on the number of tests commissioned, perhaps you only need one day. However, you must think about the time; otherwise, the test execution will have no benefit.

12.9.2 When Is It Used?

Since rapid user tests are limited to web content (or content that can be accessed via a URL), they can be primarily used when finished applications are already available online, or web prototypes can be provided. Therefore, testing products with a hardware component is not possible via this type of user testing.

Most unmoderated rapid user tests have a duration of about 20 minutes. Thus, these tests are particularly useful for less complex applications with a broad user group.

Since rapid user testing can be used whenever prototypes or applications can be deployed via the web, this kind of testing can usually be conducted relatively early in development.

12.9.3 What Do You Need?

Since the execution of rapid user tests is platform based, only a little equipment is needed for the execution of the test itself. Also, due to the limited time and testing capabilities, the material is much less extensive and complex than in a regular usability and UX test. Nevertheless, we recommend having the following materials on hand:

- A list of criteria for selecting the appropriate participants for your test (limited to the age and gender of the participants as well as a few widely used additional criteria)
- A list of the tasks that the participants are to perform on the web application (the instruction text for the participants is fundamental)
- A web-based prototype or web application that can be accessed via a URL
- A provider for rapid user testing or your own rapid user testing platform
- Time and expertise for the analysis of the test videos after the execution

12.9.4 What Are the Advantages?

The most significant advantage of a rapid user test is the speedy execution. Since you prepare all the work yourself and enter it on the platform, but everything else happens largely automatically afterward, you get the results much faster than with classic usability and UX tests.

Another advantage is the significantly lower cost of conducting rapid user tests when compared to conventional tests. Everything happens automatically, and neither external moderation nor external evaluation is involved, which means that the costs on the provider side can be kept extremely low.

Another advantage is that URL-based testing theoretically allows participants worldwide to take part in the tests quickly and easily. Whether this approach works depends, of course, on the provider and its participant database.

12.9.5 What Are the Disadvantages?

As you can imagine, however, this concept has some disadvantages. On the one hand, these tests are much less flexible than conventional tests since everything is mapped automatically via a platform, which also limits rapid user tests to content that can be provided via a URL.

Due to the lack of moderation, reacting to spontaneous events and statements is impossible, so less information is retained at some points than might have been possible in a guided test. In addition, the test duration is usually significantly shorter than in other usability and UX tests—especially because the quality of the results drops significantly when participants have to conduct longer sessions on their own. Although the *platforms* offer a fairly broad user base, the selection of participants in this context is quite generic and based exclusively on general criteria such as age and gender. Additional criteria have to be paid for, and the speed is reduced, which counteracts the idea of rapid user testing. In addition, you should assume that primarily people with an above-average affinity for technology register as testers on an online-only platform, which could distort the results of the tests.

In addition, as a customer, you need to bring a great deal of methodological knowledge to a rapid user test, perhaps even more than would be the case with a conventional usability and UX test. Why? Because you must create a good test setup, even though you'll neither be present nor able to intervene in what is happening. Thus, you must prepare everything as well as possible in advance to ensure you get the answers to your questions at the end. Unfortunately, if something goes wrong now, you'll only find out at the end (due to how the tests work) when you have already received the finished videos. In the worst case, it's back to square one.

12.9.6 What Are the Alternatives?

If the speed of execution is your concern, then moderated usability and UX sprints (Section 12.8) can be similarly fast. However, a classic usability and UX test (see XXX) can be used if you're concerned with observation. There are a few alternatives if you have decided to conduct a rapid user test because of the cost. A usability and UX sprint is similarly fast. Still, since you have people on-site and moderation and evaluation, the costs usually exceed those of a rapid user test. Only online questionnaires are similarly inexpensive (see Chapter 9, Section 9.3.5, and Section 12.13) as is the analysis of a website's click data (see Chapter 9, Section 9.3.8 and Section 12.15), but these methods provide a rather different type of data. Online questionnaires work only for surveys about opinions and do not provide data on whether something works or not. Click data can only be viewed for live systems but does not provide any assistance in interpreting the data.

Table 12.3 shows the profile for this method.

Profile: Rapid User Test	
Brief description	Rapid user testing is a method for the fast and automated testing of content that can be provided via a website. The tests are presented as a mixture of written tasks and questions, which the users navigate themselves.

Table 12.3 Profile for the rapid user test method.

Profile: Rapid User Test	
Advantages	■ Speedy implementation. ■ Cost effective. ■ Low effort for the execution as the execution is automated. ■ Easy to use worldwide.
Disadvantages	■ Since the tests are unmoderated, you cannot intervene or follow up. ■ The test quality is highly dependent on methodological competence and the guideline since there no changes can be made during the execution. ■ Recruiting user groups with specific characteristics (e.g., an occupation) is difficult or impossible—and if possible, a time disadvantage arises. ■ The scope of the tests is quite limited in time, as participants who are on their own are less motivated to take long tests. ■ Rapid user tests are only possible for content that is accessible online.

Table 12.3 Profile for the rapid user test method. (Cont.)

12.10 A/B Testing

A/B testing is a unique form of usability and UX testing in which at least two solution variants are compared. This testing can be conducted either with prototypes or in live environments.

In the first case, A/B testing corresponds to a test variant of usability and UX testing, where two prototype versions are compared. In the second variant, the implementation works differently. To present the method in this section, we focus on A/B testing in live systems (Stegemann & Suwelack, 2020; Siroker & Koomen, 2013).

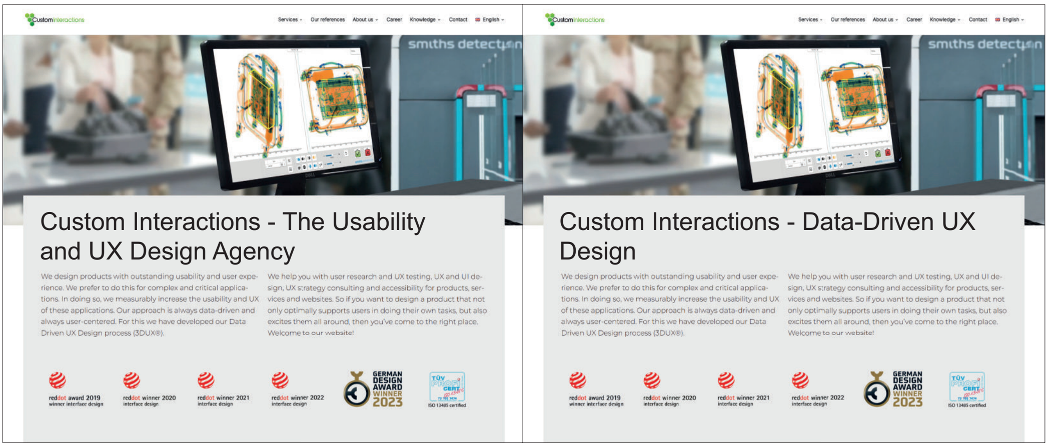
12.10.1 How Does It Work?

A/B tests are mainly tool based. Whether A/B testing is possible depends heavily on the technical platform on which your website or online store is hosted. If your website runs on WordPress, which is the case for about 40% of all websites worldwide, then A/B testing is possible almost without any problems. All you need to do is install a suitable plug-in. The same advice applies to large online store providers.

But let's leave the technology aside for now. The idea of A/B testing live applications is not so much to compare two or more completely different solutions as to fine-tune existing ones.

Meaningful A/B testing on a live system requires excellent planning and a sensible and structured approach. Since everything you test here is played out to your users, the things you want to test must be largely ready. Usually, the users are unaware they are contributing to an A/B test with their activity. Still, they are simply shopping in a web store, for example, or visiting a website to get information. So, these A/B tests only work with elaborated ideas—or you accept the damage to the brand's image. An alternative is to test still immature ideas with only a small group of users. This approach limits the damage to just a few people but, unfortunately, only works if you already have a sufficiently large user group.

For an A/B test, you usually proceed by deciding which detail you want to change on the web page. For example, that could be a headline, a button label, or a contact information position. Then, you set up an A/B test where you keep everything the same—except for just that one thing. Thus, you end up with two versions of the same web page, for example, one of which uses Headline A and the other of which uses Headline B. Everything else is the same, as shown in Figure 12.9.


CustomInteractions
Services
Our references
About us
Career
Knowledge
Contact
English
smiths detection
reddot award 2019 winner interface design
reddot winner 2020 interface design
reddot winner 2021 interface design
reddot winner 2022 interface design
GERMAN DESIGN AWARD WINNER 2023
TÜV PROFI CERT ISO 13485
ISO 13485 certified

Custom Interactions - The Usability and UX Design Agency

We design products with outstanding usability and user experience. We prefer to do this for complex and critical applications. In doing so, we measurably increase the usability and UX of these applications. Our approach is always data-driven and always user-centered. For this we have developed our Data Driven UX Design process (3DUX®).

We help you with user research and UX testing, UX and UI design, UX strategy consulting and accessibility for products, services and websites. So if you want to design a product that not only optimally supports users in doing their own tasks, but also excites them all around, then you've come to the right place. Welcome to our website!

Custom Interactions - Data-Driven UX Design

We design products with outstanding usability and user experience. We prefer to do this for complex and critical applications. In doing so, we measurably increase the usability and UX of these applications. Our approach is always data-driven and always user-centered. For this we have developed our Data Driven UX Design process (3DUX®).

We help you with user research and UX testing, UX and UI design, UX strategy consulting and accessibility for products, services and websites. So if you want to design a product that not only optimally supports users in doing their own tasks, but also excites them all around, then you've come to the right place. Welcome to our website!

Figure 12.9 Our website during A/B testing. Have you discovered the difference yet?

Now, start the A/B test. Depending on the tool, you can define the end of the A/B test by time ("User test runs for a week until Sunday noon") or by the number of people who have seen the different versions. Some tools also offer an automatic stop when enough information has been gathered to prompt you to make a decision. The tool automatically ensures that half of the visitors see one version and the other half see the second.

The results from such an A/B test are usually purely quantitative and must be defined and created by you in advance. For example, *conversions* are usually measured. For this task, you can define in advance what counts as a conversion: In the case of a store, a conversion could be making a purchase; in the case of a website, it could be subscribing to a newsletter.

Afterward, you can check in the tool to see which version performs better in terms of conversions. As with anything you do, it's imperative with A/B testing to think about the right *metrics* in advance. You can choose just one, such as *conversion rate*. But looking at different metrics and considering them together, such as the number of purchases made and the average purchase volume, can be helpful.

Once you can determine a winner based on your measurement data, the A/B test is completed. Usually, the “winner” of the test is made live for all users, so from then on, only one version (the better one) is available for everyone again.

For this procedure to work well, a crucial consideration is that you only change one thing per test run. If you turn many screws simultaneously, you won’t be able to tell at the end which change was effective. For example, you could theoretically change the headline, the button label, and the position of the contact button at the same time and find out that the conversion rate has dropped by 3%. This result would make you discard the experiment and stick with the old version. But maybe the different headline led to a 2% better conversion rate; the different button text also led to a 2% better conversion. However, the new position of the contact button led to 7% less conversion because users suddenly saw your page as less trustworthy. Then, you would have just given away 4% conversion due to a lousy test setup.

Therefore, A/B tests on live systems are usually planned rigorously and implemented on a small scale. In return, the execution time can be extremely short, at least if you have sufficient website or store visitors. In most cases, A/B tests are permanently used so that tests are constantly taking place somewhere. For this purpose, many small tests are often run one after the other. Substantial providers, such as Amazon, rely on continuous testing and improvement. Recently, Amazon even offered this feature to its retailers and affiliates so that they can place and sell their own products even better.

12.10.2 When Is It Used?

A/B tests are usually used mainly on websites and online stores or in advertising on social media that regularly have a larger number of visitors. The type of A/B testing described in this section is thus more of an optimization activity than a creative activity, which makes A/B tests particularly suitable for existing products in live operation. Conventional usability and UX tests that present the different variants to the participants are suitable for comparative testing of different variants in the prototype or concept stage.

12.10.3 What Do You Need?

Since A/B testing is implemented technically, the requirements are relatively clearly defined. Therefore, for the implementation, you need the technical possibility to play out at least two variants to users and measure their performance. As a result, you’ll need the following materials:

- A tool that supports A/B testing for your application
- At least two variants that you want to compare (and that are suitable for use in a live system)
- A definition of success
- A sufficiently large number of users using your live systems

12.10.4 What Are the Advantages?

Advocates of A/B testing argue that A/B tests on live systems deliver much better results than usability and UX tests because they do not simulate the behavior of users but instead capture and compare actual behavior. Thus, measurable differences are easily tangible and visible. And they are right about that. However, to defend usability and UX tests, note that these tests are used in development and thus pursue a different objective. Especially in the case of high-traffic web offers, the execution of an A/B test can be speedy. Thus, in some cases, an A/B test can be carried out and evaluated faster than it takes to discuss which change should now be tried.

In addition, the procedure for A/B tests is structured. The fact that only individual aspects are adjusted in a targeted manner means that the learning effect is quite high. Theoretically, knowledge can be acquired and applied to other websites and online stores. Permanent and step-by-step improvement can be driven forward through rigorous testing with large numbers.

12.10.5 What Are the Disadvantages?

Because they are used on the live system, A/B tests are rather systematic and small scale. Thus, they are less suitable for covering giant leaps or changes: Especially if entirely new and innovative concepts are to be used, direct use in an A/B test is relatively risky because actual revenue losses are expected if a concept does not work and only finds little acceptance.

So, if A/B testing is used as the only method, then further development can be expected to take the form of an evolution rather than a revolution, and thus, potential may be wasted in implementation. If A/B testing is also used for more extensive tests, bad test variants can fall back on the company. Therefore, a little intuition is required at this point in the selection and preparation—especially because people do not know it is a test.

Another disadvantage of A/B testing is the complete lack of subjective data to help justify the differences in results. As a result, you learn what works and what doesn't step-by-step, but for the most part, you must rely on a trial-and-error approach because other than the fact that something worked or didn't work, there is no evidence of improvement. Thus, you are always guessing about the "why."

12.10.6 What Are the Alternatives?

Using usability and UX tests (Section 12.7) provides a first impression of two variants in comparison. Theoretically, rapid user tests (Section 12.9) could be performed using a copied instance of the existing website. If an adjustment of the visual appearance should be made and queried, online surveys (see Chapter 9, Section 9.3.5) can also be used, whereby the focus is on determining subjective data.

Table 12.4 shows the profile for this method.

Profile: A/B Test	
Brief description	A/B testing is a unique form of usability and UX testing in which not one prototype or solution is tested in-depth, but at least two differing variants or prototypes are compared. This testing of variants is about fine-tuning individual elements, where one group of users is shown the “live” solution *variant 1,* and another group is shown *variant 2*. Participation in live A/B testing is unconscious for most users.
Advantages	▪ Enables objective statement about the better variant. ▪ When testing prototypes, includes all the advantages of classic usability and UX tests. ▪ In live tests, the decision is based not on simulated data but on actual usage data. ▪ Enables rapid hypothesis testing and structured UI testing and revision.
Disadvantages	▪ For prototype comparisons, same disadvantages of usability and UX tests. ▪ In live testing, allows mainly testing of minor adjustments, whereas large development leaps are challenging to test. ▪ Does not provide subjective data to explain user behavior.

Table 12.4 Profile for the A/B testing method.

12.11 UX Benchmarking

UX benchmarking is a form of usability and UX testing conducted as a mixture of task observation and survey. In contrast to the classic usability and UX test, however, the main aim is to classify and measurably compare a product or a product variant in comparison with other products or product variants. In other words, this method focuses on evaluating one’s own product in relation to others (Moran, 2020).

12.11.1 How Does It Work?

UX benchmarking tests can be conducted either with their own product and some competitors, exclusively with competitors without their own product, or with different variants of their own products over time. Since UX benchmarking is primarily used to justify development decisions and prove development success, the number of participants for a UX benchmarking test is usually chosen to be significantly higher (approximately 80 to 100 people) than for a classic usability and UX test. However, the structure itself is relatively comparable.

In the beginning, the *user group* that is to be addressed with the product is also defined for UX benchmarking. Criteria are derived that make it possible to identify suitable participants for UX benchmarking, and these criteria are recorded in a screener.

The next step is to determine which type of benchmark should be performed. Once you have decided on the type of UX benchmark, you'll also know how many comparisons need to be performed, allowing you to start planning the UX benchmark activity. Since the goal is to benchmark, you should develop a *test guide* whose tasks cover the core tasks of the application and do not target special cases or mini features.

The follow-up questions to the tasks you include in your UX benchmarking test should also be standardized and generally valid. In this context, you can also fall back on *standardized test procedures* (described with an example in Section 12.13). An essential step is that you change little to nothing in the surveys (also in the future) so that you can repeat the UX benchmarking later and relate it to previous tests.

After the test guide is completed, UX benchmarking is conducted. This process involves taking people through a moderated test in person or remotely. The test usually consists of tasks followed by a follow-up survey. This procedure is repeated for all websites, online stores, or variants involved in the test. The order is not the same for all participants but is modified to ensure that the order has little or no influence on the results of the UX benchmarking. This premise is usually achieved by having one part of the participants start with solution A, then see solution B, then solution C, then solution D, and then solution E. Another part starts with solution B, then solution C, then solution D, then E, and only then with A and so on, as shown in Figure 12.10.

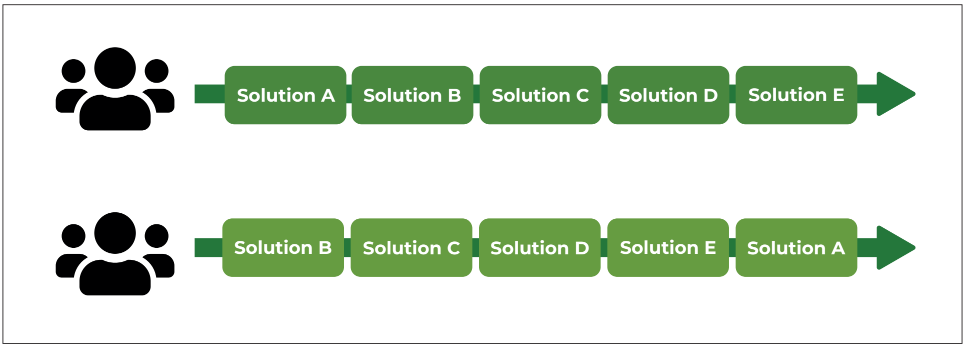


Figure 12.10 Test rotation in UX benchmarking to avoid sequence effects (e.g., "B performs better because everyone used A before").

12.11.2 When Is It Used?

For UX benchmarking to make sense, the product to be benchmarked must be a version that is accessible to you, which is usually the case when websites or online stores must be compared. Performing UX benchmarking becomes more difficult for other applications, as you need access to all market companions. But that shouldn't be a dealbreaker—it just increases the effort. For example, in the automotive industry, entire divisions do nothing but take apart and analyze competitors' finished vehicles. In this context, the focus is often more on technology, but UX benchmarking would be conceivable without further ado. By the way, the whole thing doesn't necessarily have the whiff of industrial espionage that might have come to your mind. We have seen in practice that car manufacturers pragmatically swap models on the market because it is clear that this will happen anyway.

Once you get to the competitor's products and have decided on UX benchmarking, it's just a matter of deciding what *type of UX benchmarking* you should do, according to the following considerations:

- If you're just starting a new development and do not yet have your own solution, a UX benchmark is only suitable among the relevant competitors. In this case, you can find out during the test how appealing the respective competitors are for your own user group; where their strengths and weaknesses are seen; and what aspects, if any, could be a good positioning for your own product.
- If you already have your own solution, you can run a UX benchmark with competitors to find out how you're perceived by your target group compared to others and how you can differentiate yourself even better from your market competitors. Since the UX benchmark is based on a standardized structure with the same questions, you can also run this UX benchmark repeatedly to see if your position has changed over time compared to competitors.
- A final use case is that you can use the UX benchmark to prove your development success. In doing so, you compare different variants of your own product similar to a usability and UX test with an A/B setup. The difference, however, is that you're not comparing two possible development directions with each other but an as-is status with a new version or a new version with an older implementation. Thus, you can then look at your product over time and can prove that the activities in development have led to an improved UX. Or not.

12.11.3 What Do You Need?

Since UX benchmarking is, by and large, a more comprehensive version of a usability and UX test, the requirements for conducting it are essentially the same. Thus, to perform UX benchmarking, you need the following materials:

- A list of criteria for selecting the appropriate participants for your test (screener) and time for recruitment
- A methodology expert to create a UX benchmark for you that ensures few or no sequence effects occur in the benchmarking and help you design a meaningful post-survey that can be used permanently
- A test guide that specifies what tasks the users have to complete and what questions you want to ask in the follow-up survey
- Access to all variants that you want to compare with each other
- A possibility to observe the users during the use
- Documentation of the results, which makes a comparison of the different variants visible

Compared to the classic usability and UX test, UX benchmarking adds some challenges. For example, you need access to your product, competitor products, or outdated versions. Therefore, a best practice is to start planning UX benchmarking early.

Additionally, compared to traditional usability and UX testing, the *requirement for methodological knowledge* is significantly higher, so you should ensure that the person planning a UX benchmarking for you or at your site has a solid background in experimental design. In this case, a quick test is no longer enough. The time you don't invest in preparation will result in unusable results.

Due to the relatively large number of participants, you should also plan significantly more time for the implementation and evaluation. However, part of the evaluation effort is reduced again by the standardization of the test and also by the follow-up survey. A good idea is to create a template for evaluation and documentation that helps you compare all variants and products in a standardized way, as shown in Figure 12.11.

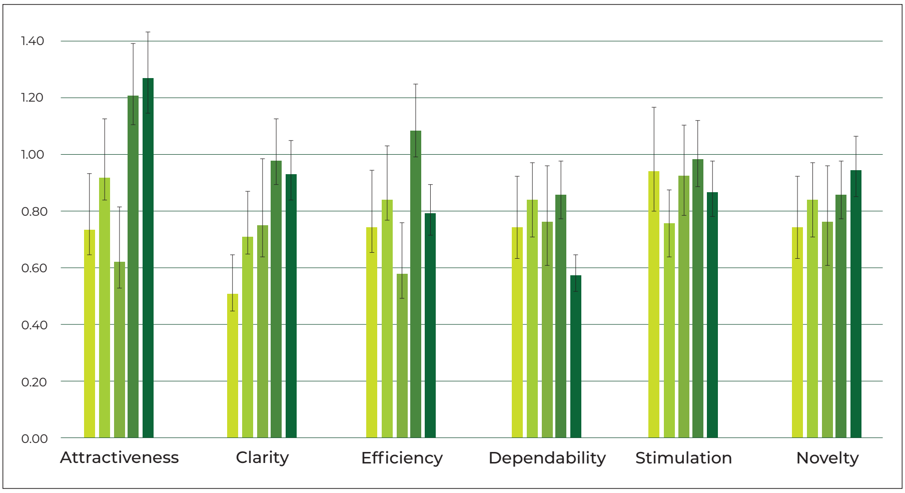


Figure 12.11 Results of a UX benchmark of the User Experience Questionnaire (UEQ) (Section 12.13.1) when comparing five products.

12.11.4 What Are the Advantages?

UX benchmarking is always beneficial if you want an objective and reliable assessment of the quality of your solution or competitor solutions. The combination of objective and subjective survey approaches, standardization, and the larger number of participants allows you to make much more reliable statements about differences between solutions than would be the case with smaller samples and less standardized procedures.

By consistently using the same structure for UX benchmarking, you can also measure changes in development and UX perception among users, thus always capturing one's own status. This capability is a huge advantage compared to all other methods that do not rely on this standardization.

Otherwise, UX benchmarking, of course, additionally offers the same advantages as a usability and UX test. In this way, you can learn to understand what works (and what doesn't) based on behavioral data. However, you also receive an additional subjective evaluation of the application and, via the comments and follow-up questions, additional hints on how the usability and ratings can be improved.

12.11.5 What Are the Disadvantages?

The disadvantages of UX benchmarking lie in the area of more complex planning. Thus, much more methodological knowledge is required to plan a UX benchmark so that no unusable results emerge.

The UX benchmark test is proportionally a costly method because the benchmarking task is conducted with many participants, and each session is moderated. On the one hand, a great deal of effort is required in preparing and recruiting the participants. On the other hand, the implementation and evaluation of many participants costs more time and money, which makes this method less suitable for young or small companies that need results quickly.

In addition, UX benchmarking relies on the fact that the operation and evaluation of different market companions can also be compared. Thus, benchmarking can only be used meaningfully if competitors' products are also accessible. This limits the application area considerably if money is not to be invested in accessing or acquiring products from market companions.

12.11.6 What Are the Alternatives?

Considered on a much smaller scale (fewer participants), a usability and UX test (Section 12.7) can serve as an alternative when comparing variants within a test. However, since fewer people usually participate in the test, you can't easily make statements about actual differences in the evaluation. Suppose the main focus is on differences in subjective evaluation. In that case, an online survey can also be used—if necessary, also with standardized questionnaires (see Chapter 9, Section 9.3.5 and Section 12.13). If the focus is on the *comprehensibility of the offer* and a *quick evaluation*, then the 5-second test (Section 12.12) can also be used as an alternative to UX benchmarking. However, insights into the solution's usability cannot be obtained with a 5-second test.

Table 12.5 shows the profile for this method.

Profile: UX Benchmarking	
Brief description	UX benchmarking is a variant of usability and UX testing in which larger numbers of users are used to objectively compare different variants (benchmarking) and select the best variant on a well-founded basis. Compared to live A/B testing, however, subjective data is generated, in addition to objective data.
Advantages	■ All the advantages of classic usability and UX testing. ■ Objective comparison between different solutions. ■ Enables a classification of one's own solution concerning its quality from the user's point of view. ■ The standardized structure of the test also allows continuous comparisons and development over time to be checked.
Disadvantages	■ All the disadvantages of classical usability and UX tests. ■ Higher costs and greater evaluation effort due to the larger number of participants. ■ When comparing to competitors' products, access to their products is necessary.

Table 12.5 Profile for the UX benchmarking method.

12.12 5-Second Test

The 5-second test is a method that comes from marketing and is designed to measure the user's evaluation of a product, advertisement, or website after only a very brief impression of 5 seconds. This test aims to measure the first impression a product triggers (Taron, Pfeiffer, & Thomaschewski, 2016). After all, you don't get a second chance to make a first impression!

12.12.1 How Does It Work?

The preparation of a 5-second test is similar to other user-based test procedures: The first task is to determine who will use your product later on and define how to recognize and attract them to participate in your 5-second test.

Since the 5-second test is primarily used to check the first impression of a website, conducting a test purely remotely poses little to no problems. This approach also simplifies the recruitment of potential participants because you can use social media campaigns, for example, to reach people from the user group in a targeted manner. If your focus is not on websites but on another solution, you may have to work with on-site tests. But the procedure is analogous.

After determining who will take your test, you prepare the test with a mini-test guide. A 5-second test consists of two parts:

- From the 5-second confrontation of the users with the solution to be tested
- From the post-confrontation interview

The 5-second test is relatively short overall (but longer than 5 seconds...) and can be completed in less than 10 minutes per participant. In the first part, the users are shown the solution and have 5 seconds to look at it. After 5 seconds, the solution is covered, closed, or hidden so that no further information or impressions can be gathered. Questions are then asked of the participants to find out what impression the users have of the solution and how well the solution can present what is on offer in a way that can be quickly understood. Typical questions that could be asked include the following:

- What is this website, page, or screen about?
- What does the company do/offer?
- What impression did you get of the company?
- How competent do you consider the company behind this website, page, or screen?

You can decide for yourself which questions to ask. The important thing is that you can capture two types of information with a 5-second test:

- Objective data on the comprehensibility of the offer within a short period
- Subjective data on the impression left by the solution

Based on the answers to the sample question “What does the company offer?” you can objectively conclude whether the offer has been formulated in an easy-to-understand and graspable way. However, all opinions and evaluations of the participants are purely subjective.

In most cases, 5-second tests are used to obtain both data types. A comparison with competitor solutions can also provide insights into how well your solution is understood compared to others.

12.12.2 When Is It Used?

A 5-second test is always appropriate when you want to check how quickly understandable the content on a page or screen is and what first impression the solution leaves. You can either work with an existing live system or show users a visually designed website prototype. In any case, the design of the prototype must correspond to the later design. Otherwise, the results of the 5-second test cannot be used. Of course, this rule does not only apply to the main screen; a 5-second test can also be used for testing any other screen so that, for example, a call-to-action (CTA) area on a website can also be tested.

Another application of this test method is the analysis of advertising, including banners and other advertisements—theoretically in the digital area as well as in the print area. Especially in advertising, quick absorption of information is extremely important, as little time is invested in understanding ads.

12.12.3 What Do You Need?

You don't need much to conduct a 5-second test. On the one hand, you need participants from the appropriate target group and, on the other hand, the solution or the advertisement to be tested and the questions you want to ask.

Whether you conduct a 5-second test in person or remotely depends on your preference and the availability of participants and your product. Since the test is usually so short, a remote format lends itself in most cases.

In addition, digital platforms are available to support you in performing the 5-second tests and automatically hide the solution (then it is an image) after the 5 seconds. This feature, of course, makes the whole test procedure much more correct and stress free for the moderator. In case of doubt, a stopwatch can do as well. To conduct this test, you'll require the following materials:

- A list of criteria for the selection of suitable participants for your test (screener)
- A mini-test guide that specifies which questions you want to ask in the follow-up
- A visually designed version of the prototype or the actual solution or advertisement at issue
- The ability to stop the time on a second basis (stopwatch or test platform)
- Documentation of the results

12.12.4 What Are the Advantages?

The 5-second test is speedy and easy to perform. You do not need particularly long preparation, special methodological knowledge, or expensive equipment. However, the formulation of the questions should be well thought out. Otherwise, the method is ideally suited for beginners to get a first impression of the effect of their solution.

The 5-second test is quite easy to perform remotely due to its structure, so it also becomes easy to include people from different regions or even nations in the test.

Additionally, because the test is concise, it is usually easy to find people willing to participate in the 5-second tests. Alternatively, different solutions or designs can be included in the test, which extends the test duration but often keeps it within a manageable and short scope.

12.12.5 What Are the Disadvantages?

The 5-second test—due to its brevity—is, unfortunately, unable to make objective statements about the usability and UX of the solution. The only objective data obtained from this test relates to the comprehensibility of the initial content. This information is worth a lot but cannot compete in scope and depth with other usability and UX tests. As a result, the informative value is minimal.

In addition, a visually mature prototype or a finished application already in live operation is required for the implementation. Feedback based on wireframes is unsuitable for applying 5-second tests since the visual design significantly influences the steering of attention.

12.12.6 What Are the Alternatives?

The 5-second test is unique in its orientation among other usability and UX methods. A few other formats that test similar aspects. If comprehensibility is in focus, then usability and UX testing (Section 12.7) or usability and UX sprints (Section 12.8) are plausible. In these cases, however, the participants naturally will need more time to interact with the solution and should be allowed to explore the content in a calm manner. Therefore, the results are only comparable to a limited extent.

Online surveys (see Chapter 9, Section 9.3.5, and Section 12.13) can also be used to record first impressions. However, since online surveys are unmoderated, a follow-up survey with open-ended questions often produces worse results than a face-to-face survey. In addition, an online survey either does not contain pictures or only contains pictures, resulting in the solution's viewing time being significantly longer compared to that of the 5-second test. Thus, no real alternative to the 5-second test exists if your goal is to record initial impressions.

Table 12.6 shows the profile for this method.

Profile: 5-Second Test	
Brief description	In the 5-second test, users are only confronted with a part of the solution for a very short period (5 seconds) to understand what information can be gained within this short time and how the solution is evaluated.
Advantages	■ Speedy and easy implementation. ■ Little knowledge of methods is necessary, and thus, this method is suitable for beginners. ■ Quite suitable for remote execution. ■ Easy to attract participants. ■ Provides quick insights into the comprehensibility and first impressions of the solution.

Table 12.6 Profile for the 5-second test method.

Profile: 5-Second Test	
Disadvantages	■ No objective data on usability and UX of the solution in use; limited to first impression only. ■ The scope of the test is severely limited due to the methodology and thus less informative than the more comprehensive methods.

Table 12.6 Profile for the 5-second test method. (Cont.)

12.13 Standardized Questionnaires

Standardized questionnaires are not a single method but a group of methods in which prepared questions are used online or in-person to capture users' opinions about a system, product, or service (Schrepp, 2019). Because the collection is standardized, a comparison with other systems becomes possible. Since it is impossible for us to describe all the procedures here, we'll first explain the general procedure for dealing with standardized questionnaires and then pick out a few examples, which we'll present in more detail.

12.13.1 How Does It Work?

The procedure for standardized questionnaires is always the same. Usually, a set of questions is developed based on an objective, which is then tested on different target groups and revised until exactly the questions have been found that have the following characteristics:

- They really measure what needs to be measured.
- They are sensitive to differences.
- They are easy to understand and use.

The creation of standardized questionnaires is usually the task of research. If you design questions that you can work well with and use repeatedly, then that is a *set of questions*, which is a kind of standardization. However, a *standardized questionnaire* only exists if your set of questions meets scientific quality criteria for the design of tests.

If you're working with standardized procedures, follow these steps: First, you must decide which *procedure* you want to use. In essence, selecting the appropriate procedure is mainly related to your question. Once you have decided on which procedure to use, you must plan the use of the procedure.

In addition, even with a standardized procedure, you need to consider which user group you would like to address and define criteria that you can use to identify these people for your survey.

Of course, since all of the processes we're about to introduce revolve around evaluating a product, system, or service in some way, you need the product, system, or service—either real or prototype—to be evaluated.

Next, you should create your test guide. You should consider how the standardized questionnaire should be integrated at this point. Usually, standardized questionnaires do not stand alone but are part of a more extensive test, for instance, a usability and UX test, UX benchmarking, an online survey, or another test procedure in which you work with users.

When planning the deployment, an essential step is to remember that, although you can ask additional questions about the standardized tool, you usually cannot deploy a standardized questionnaire in extracts or only in parts. If you do so, on the one hand, the evaluation will no longer work, and on the other hand, it won't be easy to compare it with other systems, products, or services that have also used the tool. In addition, all quality criteria are designed in such a way that the quality of the standardized test can only be guaranteed if it is used precisely in the form in which it was developed. Therefore, our tip is to use the procedures exactly as they were developed and follow the instructions for evaluating the questionnaires if you want to benefit from the advantages of standardization.

But now, we would like to introduce you to some standardized questionnaires you can use for your tests and surveys.

Pragmatic versus Hedonic Quality

Before we get started with the standardized questionnaire procedures, we would like to introduce the terms *pragmatic* and *hedonic* at this point since most of the presented procedures rely on this distinction.

The Cambridge Dictionary defines *pragmatic* as something that is "based on practical judgment rather than principles." In other words, being pragmatic has an apparent, factual reference (*https://dictionary.cambridge.org/dictionary/english/pragmatic*). A product has a high pragmatic quality if it supports the user's actions effectively and efficiently—in other words, if it has high usability.

A *hedonic* quality, on the other hand, describes all aspects of a system, product, or service beyond mere usefulness (*www.usabilityfirst.com/glossary/hedonic-quality/*). The system, product, or service is fun for the users or is perceived as particularly stimulating. Accordingly, the evaluation of product attractiveness is often made up of both pragmatic and hedonic aspects.

System Usability Scale (SUS)

The *System Usability Scale* (SUS) is probably one of the oldest standardized methods for assessing subjective usability. Developed by John Brooke back in 1996, since then, the

SUS has become one of the standards in the industry when looking for a reliable measure that can be used comparatively in many studies.

The charm of the SUS lies in its brevity and simplicity. The questionnaire consists of ten statements, answered on a 5-point scale. Each statement can be answered with a value from 0 for "do not agree at all" to 4 for "agree completely." Statements in the SUS include the following examples:

- I think that I would like to use this system frequently.
- I found the system to be unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.

The *evaluation* of the SUS turns out to be a bit complicated. Half of all statements are formulated in such a way that agreement speaks for good usability; the other half are formulated in such a way that agreement speaks for poor usability. Therefore, the first step in the evaluation is to reverse the statements for which the agreement is poor. Thus, a 3 becomes a 1, and a 4 becomes a 0. Subsequently, the values of all statements are added together. In other words, for the 10 statements per person, you have a value between 0 and 4. Therefore, the possible SUS value range per person is 0 to 40. The result is then multiplied by 2.5 to reveal the total SUS value, which can ultimately lie between 0 and 100, where 100 would be the best possible result for a system and 0 the worst.

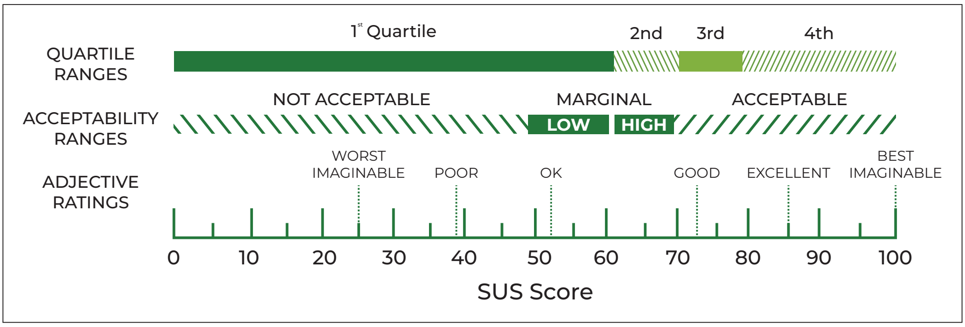


Figure 12.12 Overview of possible outcomes in SUS and their relation to other products. (Source: Adapted from Bangor, Kortum, & Miller, 2009)

The SUS is considered a "quick and dirty" method by its author, but this characterization is only partially true. Although the implementation is quick, the results have proven relatively meaningful in various studies, even with few participants, to distinguish between systems with good usability and systems with poor usability. We are not talking about minor differences of 82 to 83 points, but somewhat larger ones of 80 to 60, but the tendencies are reliable and fitting. In addition, the SUS has been used several times to perform benchmarking. In the past, tables have even been developed that provide an overview of the ranges in which a SUS result for a product is to be rated as poor, average, or rather good, as shown in Figure 12.12.

However, the SUS only provides an overview of perceived usability; other aspects (e.g., hedonic aspects) are not considered in the SUS. To compensate for this shortcoming—especially concerning the broader concept of UX—further questionnaires have been developed for use in these areas.

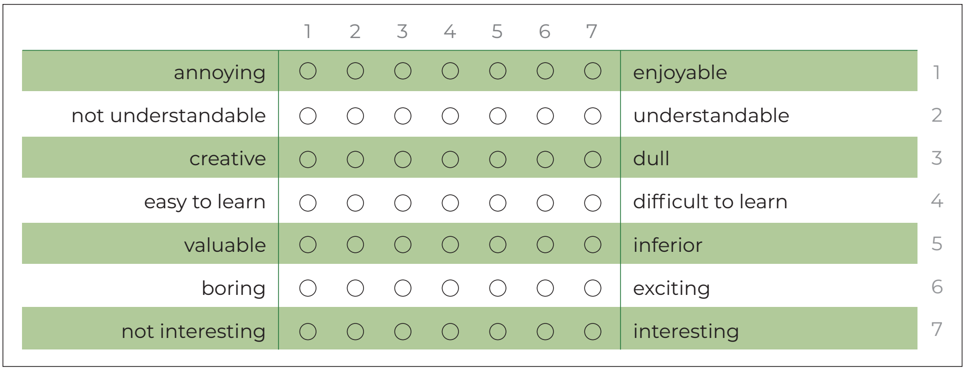
User Experience Questionnaire

The UEQ is a questionnaire that was essentially developed by Martin Schrepp back in 2005 and has been repeatedly tested and confirmed over many different studies and publications (Laugwitz, Schrepp & Held, 2008; Schrepp, Hinderks & Thomaschewski, 2014; 2017a; 2017b). If you're interested in studying the UEQ in more detail, you'll find plenty of references to related publications and the development of the questionnaire on the website *www.ueq-online.org*. However, since most people—understandably—tend not to go to great lengths to compile this information, the UEQ manual has also been available on the website since 2017 and summarizes everything you need to know.

The UEQ aims to measure the UX using a standardized procedure. Schrepp mentions the following scenarios as possible applications:

- The comparison of the UX of two different products (for example, existing version versus new version)
- To determine whether a product has a sufficient level of UX
- To identify areas for improvement

Many UX experts participated in developing the questionnaire to define the core constructs that make up the UX. The result is a questionnaire that consists of a total of 26 pairs of opposites that can be assigned to six different scales, as shown in Figure 12.13.



	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7

Figure 12.13 Example word pairs from the UEQ. (Source: www.ueq-online.org)

This way of designing questionnaires with pairs of opposites uses what are called *semantic differentials*—just in case you ever want to show off. The six scales of the UEQ include the following:

- **Attractiveness**
  What is the overall impression of the product? Do users like the product or dislike it?
- **Perspicuity**
  Is it easy to become familiar with the product? Is it easy to learn how to use the product?
- **Efficiency**
  Can users solve their tasks without unnecessary effort? Does it react fast?
- **Dependability**
  Does the user feel in control of the interaction? Is it secure and predictable?
- **Stimulation**
  Is it exciting and motivating to use the product? Is it fun to use?
- **Novelty**
  Is the design of the product creative? Does it catch the interest of users?

All scales except the attractiveness scale are measured using four pairs of opposites. The *attractiveness* scale is measured using six pairs of opposites and is a pure *valence scale*. Thus, the scale only assesses the perceived valence of the product. The *perspicuity*, *efficiency*, and *dependability* scales measure the pragmatic quality of the product, whereas the *stimulation* and *novelty* scales measure the hedonic quality of the product.

To ensure that the UEQ can also be used as a benchmark, the questionnaire has been used in many studies and has been tested on various products, which has resulted in comparative values into which a company's products can be sorted. Almost 10,000 people have evaluated a total of 246 products. The result is a diagram with benchmark values with which your own product can be compared, as shown in Figure 12.14.

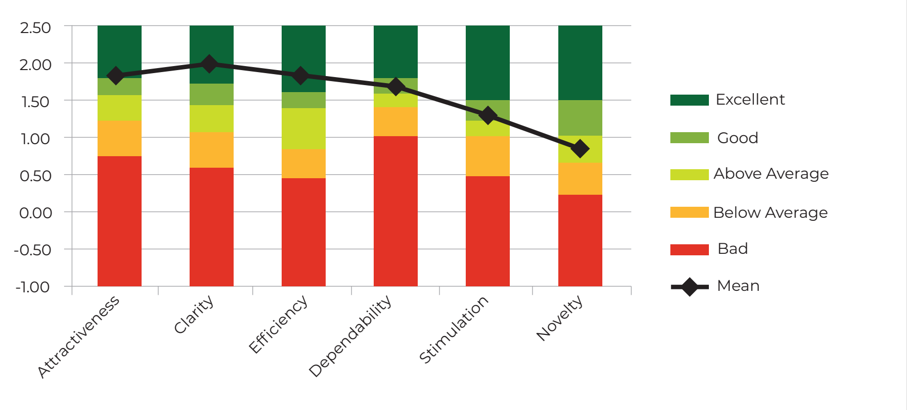


Figure 12.14 Benchmarking graph for the classification of test results into comparative values at UEQ. (Source: www.ueq-online.org)

This benchmarking type is necessary because the full scale is usually not used to evaluate products. In fact, the UEQ ranges from +3 to –3 on each scale, but rarely will a product actually reach these extreme values, so the range in which products are evaluated in practice is much narrower.

To simplify the evaluation, Schrepp provides a Microsoft Excel spreadsheet that you can download from *www.ueq-online.org*. This spreadsheet also explains in detail how the data you obtain can be used. A short version of the UEQ also works with fewer pairs of opposites and enables a faster assessment than the complete UEQ. However, Schrepp only sees a few minimal application scenarios for the short version, which does not work as well as the long version. The use of the short version is recommended for the following scenarios:

- To survey visitors on web stores or websites when they leave the page, when there is little time
- For integrating the short version into an existing survey instrument, which should not be significantly prolonged by it
- For the comparison of many different products or product variants

The short version has only eight pairs of opposites: four measuring pragmatic quality and the remaining four measuring hedonic quality. However, Schrepp again points out that comparing the scale values for different products in this context is difficult because the pragmatic and hedonic qualities do not have the same relative importance for every product.

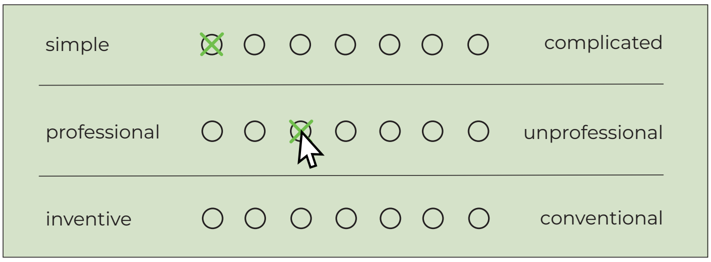
AttrakDiff

Another standardized method that assesses the subjective usability and appearance of the interactive and appearance of your interactive product from the users' point of view is the *AttrakDiff* (Hassenzahl, Burmester & Koller, 2003; 2008; Burmester, Hassenzahl, & Koller, 2007). On the one hand, the AttrakDiff measures the perceived attractiveness of your product. On the other hand, this method also measures the perceived pragmatic and hedonic quality from the users' perspective.

What is the beauty of AttrakDiff? The questionnaire is free of charge and available as an interactive tool at *https://www.attrakdiff.de/index-en.html*. AttrakDiff can be easily used there by anyone—as long as the input does not exceed 20 people. The use purposes for AttrakDiff include the following scenarios:

- The one-time evaluation of a product
- A comparison of two products
- A comparison of a product before and after

The questionnaire has a long history and was jointly developed in 2003 by Marc Hassenzahl, Michael Burmester, and Franz Koller. Through the AttrakDiff method, a product is evaluated based on 28 pairs of opposites rated on a 7-point scale, as shown in Figure 12.15.



simple ☒ ○ ○ ○ ○ ○ ○ complicated

professional ○ ○ ☒ ○ ○ ○ ○ unprofessional

inventive ○ ○ ○ ○ ○ ○ ○ conventional

Figure 12.15 Example word pairs like those offered by AttrakDiff (https://www.attrakdiff.de/index-en.html).

The online tool makes evaluating the questionnaire easy, as the results are entered in an easy-to-understand matrix. As a user, you can then use this matrix to see how your product is rated by the users, as shown in Figure 12.16.

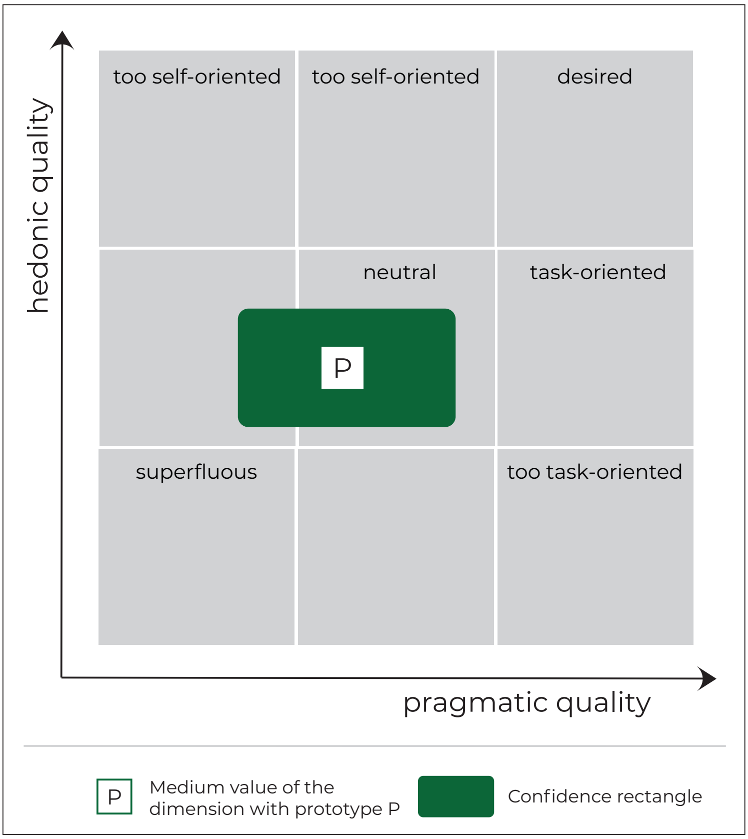

hedonic quality
too self-oriented
too self-oriented
desired
neutral
task-oriented
P
superfluous
too task-oriented
pragmatic quality
P Medium value of the dimension with prototype P
Confidence rectangle

Figure 12.16 Example evaluation matrix for a product similar to those offered by AttrackDiff (https://www.attrakdiff.de/index-en.html).

The small box with the P shows the average assessment in the evaluation, and the larger surrounding box shows how uniform or inconsistent the assessments were between the different people. In this specific example, most users would have agreed that the tested product has a high hedonic quality. In the case of the pragmatic quality, however, the users' opinions would have been significantly more varied, as seen from the wider green box. The labels of the boxes give you an indication of how the product is to be rated overall.

If you're comparatively testing a product, you can see a visual comparison of the two variants side by side via the tool provided. You can quickly identify which product performs well in which dimension, as shown in Figure 12.17.

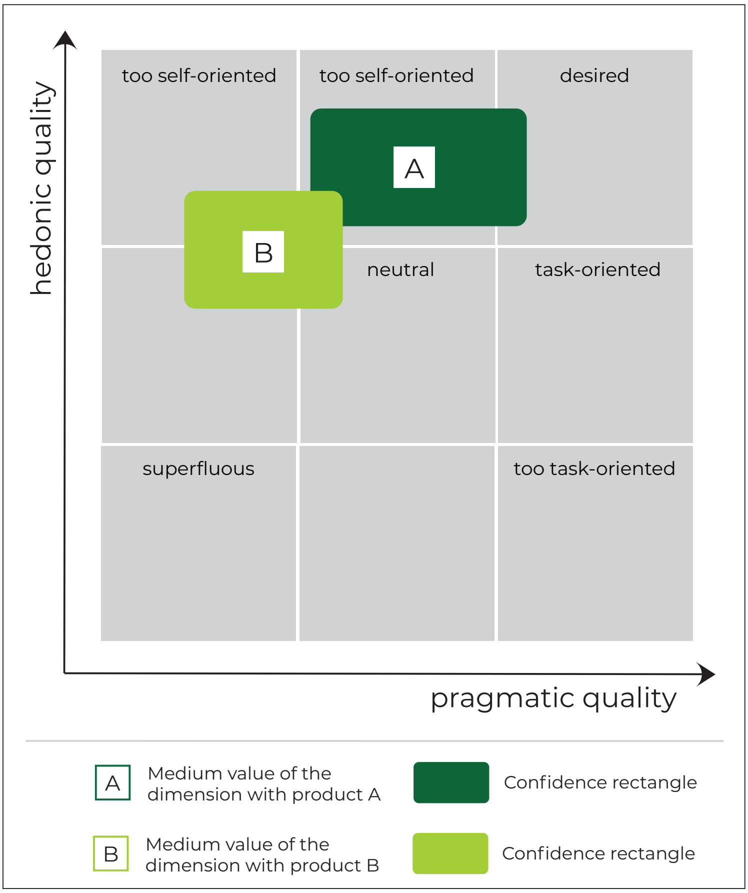

hedonic quality
too self-oriented
too self-oriented
desired
A
B
neutral
task-oriented
superfluous
too task-oriented
pragmatic quality
A Medium value of the dimension with product A
Confidence rectangle
B Medium value of the dimension with product B
Confidence rectangle

Figure 12.17 Example evaluation matrix for product comparisons similar to those offered by AttrackDiff (https://www.attrakdiff.de/index-en.html).

INTUI

Compared with the SUS, the INTUI is a relatively new survey instrument developed jointly by Daniel Ullrich and Sarah Diefenbach starting in 2010. INTUI focuses on a particular component of UX, namely, the intuitiveness of a product (Ullrich & Diefenbach, 2010; Ullrich, 2013; 2014). For a long time, one of the key questions in good product design has always been: “Can you use the product intuitively?”

In this context, INTUI comes in and measures intuitiveness not only based on a single question but on a holistic questionnaire that covers various components of intuitiveness. According to Ullrich and Diefenbach (*www.intuitiveinteraction.net*), intuitiveness consists of the following four core components:

- Gut feeling
- Verbalizability
- Effortlessness
- Magical experience

Gut feeling means that users usually describe an intuitive system so that the operation is not analytical and deliberate but unconscious and guided by feelings. This feeling is also reflected in users that find it difficult to explicitly name the steps that must be implemented for a successful operation (*verbalizability*). Behind this idea is that learning and action are mainly implicit in intuitive systems, and implicit knowledge is much more difficult to convey than explicit knowledge. In addition, interaction with intuitive systems often feels quick and easy for the user (*effortlessness*). If no difficulties or stumbling blocks get in the way, operations feel easy.

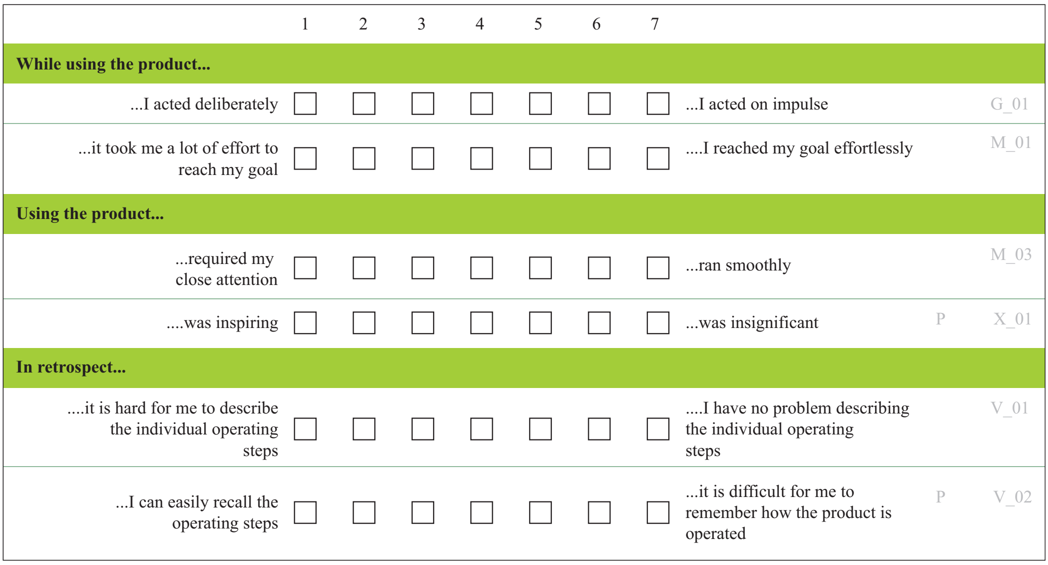
The last dimension, the *magical experience*, arises from the users’ inability to properly assess their own contribution to the simple interaction—namely, their implicitly accumulated knowledge—and from the fact that a system that works so simply, effortlessly, and without much thinking triggers genuine enthusiasm. This enthusiasm that users feel about a system’s good design is the part that can feel magical.

These four dimensions of intuitiveness are measured by the INTUI in a standardized manner using 16 statements. All statements are rated on a 7-point scale, at the end of which each has an answer. Depending on how the users answer the statement, they tend to agree with one or the other or with neither or both statements to the same extent.

The INTUI method divides these statements into three different areas, as shown in Figure 12.18:

- Statements about the perception of one’s behavior while using the product
- Statements about the perception of the use of the product itself
- Statements about own perception of usage in retrospect usage

For the evaluation, the two developers of the tool also provide an *SPSS* file (SPSS is statistical software from IBM), which helps you use the individual values correctly and helps you evaluate the questionnaire. With the help of the INTUI method and the evaluation table, you can make a standardized evaluation of intuitiveness easily and also compare this value again between different systems. Unfortunately, we do not know any benchmark values now, so you can use the questionnaires primarily for comparative purposes.



	1	2	3	4	5	6	7			
While using the product...										
...I acted deliberately	☐	☐	☐	☐	☐	☐	☐	...I acted on impulse		G_01
...it took me a lot of effort to reach my goal	☐	☐	☐	☐	☐	☐	☐	....I reached my goal effortlessly		M_01
Using the product...										
...required my close attention	☐	☐	☐	☐	☐	☐	☐	...ran smoothly		M_03
....was inspiring	☐	☐	☐	☐	☐	☐	☐	...was insignificant	P	X_01
In retrospect...										
....it is hard for me to describe the individual operating steps	☐	☐	☐	☐	☐	☐	☐	....I have no problem describing the individual operating steps		V_01
...I can easily recall the operating steps	☐	☐	☐	☐	☐	☐	☐	...it is difficult for me to remember how the product is operated	P	V_02

Figure 12.18 Example statements in the INTUI method. (Source: http://intuitiveinteraction.net)

12.13.2 When Is It Used?

We've covered, in rudimentary form, the individual methods themselves, but we would like to list the purpose of standardized questionnaires for recording UX again here. Usually, you use standardized questionnaires whenever you want to do the following activities:

- Compare a system, product, or service
- Compare a system, product, or service in different development stages or versions
- Understand how your system, product, or service generally compares to others

Standardized methods always develop their full potential when a comparison group or comparable values exist. Using standardized methods without comparison values—if no reference values are available from other studies—is usually not that useful since users always refer to a reference value to make an assessment. If you and your test do not give this reference value, then a frame of reference is established by the users themselves based on previous experience with different products, which is not desirable.

In general, you always use standardized procedures only downstream of an exploration. Therefore, if you plan a test in which the participants use the product and solve tasks with it, then use the standardized procedures to quantify the experience following this use.

12.13.3 What Do You Need?

The use of standardized questionnaires is relatively straightforward in most cases. For this method, you'll need the following materials:

- Access to the standardized procedure you want to use
- Your system, product, or service (or a prototype) to be tested
- The possibility of delivering your solution and the questionnaire to the appropriate user group—either in person or digitally
- A setup that allows everyone to have comparable contact with your solution
- An evaluation scheme for the respective procedure

Standardized procedures are often not used individually but as part of a more extensive test in which, for example, usability and UX are also tested based on predefined tasks. Then, of course, you also need everything that belongs to the structure of the test.

12.13.4 What Are the Advantages?

Properly used, standardized procedures bring with them the advantage that they enable comparability of products that is otherwise not possible or only possible in part. Thus, you can more easily assess the status of your product compared to competitors and get a better sense of where you stand. This understanding is much more challenging to achieve with non-standardized evaluation procedures, as these are rarely used comparably across multiple tests or studies.

However, you can just as easily use standardized test procedures to track the progress of your products. For this task, you can apply the standardized procedure to your product regularly and thus obtain comparative values that make the quality of the product and the quality growth visible. When standardized questionnaires are used as a standalone method, they can usually be provided online to a large number of participants at the same time. This approach usually allows the test to be administered relatively inexpensively and quickly. In addition, the online provision also enables worldwide use, provided the questionnaires are available in different languages.

12.13.5 What Are the Disadvantages?

Most standardized procedures are scientifically elaborately constructed and based on a precise model of how usability or UX is composed. On the one hand, this precision is good, but on the other hand, the quality of the standardized questionnaire depends on whether you use the questionnaire exactly as intended. Deviations are practically impossible without reducing the comparability and informative value of the procedure. As a result, you lose flexibility.

Furthermore, many standardized procedures are relatively long and extensive due to their highly scientific quality, so using the tools has already become a time factor in implementing the studies that should not be neglected. You may have to plan 20 minutes to complete the questionnaire to compare three products. However, some methods counter this problem by developing short versions of the questionnaires to reduce the time required significantly.

12.13.6 What Are the Alternatives?

For comparability with other products, on the one hand, you can use UX benchmarking (Section 12.11), in which the same structure is used again and again for testing the different products.

On the other hand, if you're concerned solely with the internal comparability of different solutions, you can develop *your own set of questions* that you can then use for evaluation. Before you go about this, however, you should be sure that you're asking questions that will subsequently measure what you want to measure. Again, validity is the issue: If you measure size with a thermometer, the result will always be the same but will never make sense.

Table 12.7 shows the profile for this method.

Profile: Standardized Questionnaires	
Brief description	Standardized questionnaires represent a group of survey methods in which users are asked to evaluate a system, product, or service in written form. The evaluation usually takes place using scales and closed questions, whereby the survey's focus depends on the questionnaire used.
Advantages	■ This method enables the comparison of different solutions with each other and the assessment of the quality of one's solution from the user's point of view. ■ Fast and cost-effective testing. ■ Implementation is possible with many people at the same time. ■ Well suited for remote testing and thus also for worldwide use.
Disadvantages	■ No adaptation of questionnaires is possible if comparability is to be maintained. ■ Most standardized questionnaires are relatively large, making it difficult to compare many variants.

Table 12.7 Profile for the standardized questionnaires method.

12.14 Analysis of Quantitative Usage Data

The *analysis of quantitative usage data* refers to the use of all data generated by existing systems, products, or services (cf. Schmidt-Thieme & Gaul, 2002).

In most cases, this data includes data on the frequency or duration of use of certain features. However, you may have information about support requests or feature requests via the support hotline. The latter scenario is the focus of our discussion. Therefore, in this section, we're focused not on creating new data, but on using existing data created during the solution's use.

12.14.1 How Does It Work?

The solution in which usage data is to be analyzed must have tracking functionality to analyze quantitative user data. This tracking does not have to be user specific; it is sufficient if the functions used and the length of time over which the individual functions are used are recorded anonymously. Data collection can be automated or manual, as already described in our earlier example of the support hotline.

With the information about which functions are used, a purely quantitative analysis can be performed that provides an initial insight into the importance of the existing features from the user's point of view. Since no other data is available, the only option you have at this point is to create a ranking in which the most frequently used functions are in the first place, followed by the less frequently used functions in the subsequent places.

Note at this point that problems can already arise if you rely only on the analysis of quantitative user data. For example, a function could end up in last place because it is seen but ignored by the user (i.e., considered unnecessary). But the function could also end up in last place because users could not find it and, therefore, it is unused. We'll come back to the disadvantages later in Section 12.14.5.

A similar evaluation allows for analyzing support tickets and support requests in general. The tickets can also be analyzed and evaluated. A typical procedure for analyzing support tickets would be to categorize the tickets into different topic groups and then count their frequency so that conclusions can be drawn about areas with many problems (and areas with few problems) based on the feedback.

This step is followed by a detailed analysis of the high-priority areas to understand the users' problems. The problem descriptions can then be used to develop suggestions for improvement.

12.14.2 When Is It Used?

Quantitative usage data or customer feedback is only possible if a product is already in live operation that collects usage data, which is usually the case if a web-based tool is hosted in the cloud. With locally running applications, getting at the data is usually difficult, either because it may not be collected internally or because it is not provided externally.

Experience shows that performing a ticket analysis is generally easier. Many companies, at least medium-sized ones, have internal or external ticket systems. However, this data is rarely used systematically, so analysis can be performed relatively quickly and with high benefit.

12.14.3 What Do You Need?

Compared to other methods, there is little need for additional material in analyzing quantitative usage data. With this method, the *legal framework* is more of a challenge. Especially when collecting data from internal employees, you must ensure that the data collection complies with any company agreements and data laws like General Data Protection Regulation (GDPR) in the EU so that you don't drift into a legal discussion about recording employee performance. Once you have secured the framework for this data collection, however, the list of required materials is relatively short:

- After all, the system, product, or service is already live and used.
- A best practice is to have a tracking functionality in the tool that records which function is used (how often/how long).
- In addition or alternatively, you need a ticket system where all tickets are collected.

12.14.4 What Are the Advantages?

The biggest advantage of analyzing quantitative usage data is probably that you don't need to conduct an extra phase for data collection. Your users simply use the system, product, or service as they always have, and the analysis can be performed at any time without having to invite extra people or prepare guides and prototypes.

The second major advantage is the often sheer volume of available data. Many data points can usually be accessed, especially for solutions running for a long time. In terms of usage frequency, you can thus formulate much more reliable statements than a small usability and UX test with only six people. Since resources for revising and improving solutions are limited, choosing a prioritization that brings the most significant added value for many people is easier.

In addition, the way the data is collected makes repeated or even continuous evaluation possible. Since the data reflects the actual usage behavior of the users, this data can also be used to make reliable statements about changes in user behavior.

12.14.5 What Are the Disadvantages?

Even though the amount of data points in quantitative usage data analysis is huge, some difficulties limit the use of the data and provide only limited insight into the underlying problems. For example, while you get plenty of data on which features are used frequently in your application, there is no telling why that is the case, what is done with them, or why other features are not used. Without background knowledge, developing solutions that solve existing problems without creating new problems can be difficult.

The analysis of the tickets also reveals a problem for improving the application: Users may use tickets primarily for purely technical problems that prevent them from using their application or solving their problems but do not report other problems at all. When asked, users might be annoyed about the perceived inefficiency of the system, for example, but no ticket is submitted because “actually everything works.”

Support tickets always focus on the things that don’t work technically, whereas problems in use are rarely reported back via ticket systems. This discrepancy mainly arises because error messages are often easy to report, for example, by referring to an error code. However, usage problems are often unconscious or complex, so ticket creation takes up a user’s time and is thus unattractive to many people. Thus, over time, you may have solved all the technical problems but still have a mediocre or poor application from the user’s perspective because it simply doesn’t solve the right problems. Since few people bother writing detailed support tickets about usage issues, they will always be low priority compared to technical issues reported more frequently.

In general, analyzing quantitative usage data requires a system, product, or service that is already live and enables the collection of usage data.

12.14.6 What Are the Alternatives?

Few alternatives to the quantitative analysis of usage data exist that can provide comparable information. For a web application, then click analysis can be used that track web page usage (see Chapter 9, Section 9.3.8, and Section 12.15). All other methods provide subjective or objective usage data, but on a much smaller scale and not in the real usage environment. What works quite well is combining an analysis of quantitative usage data with a qualitative study, such as a usability and UX test (Section 12.7). This dual approach increases the required effort—but you compensate for the disadvantage of missing the “why.”

Table 12.8 shows the profile for this method.

Profile: Analysis of Quantitative Usage Data	
Brief description	Using existing data from the use of a solution to gain new insights into usage without having to collect new data.
Advantages	▪ The data reflects actual usage behavior. ▪ No effort is required for additional data collection—neither by you nor by the users. ▪ Large amounts of data are available for evaluation, so many hypotheses can often be tested. ▪ Continuous data collection makes it possible to repeat the evaluation at different times and discover changes in user behavior.
Disadvantages	▪ It only works if there is already a solution/predecessor solution. ▪ Poor understanding of the "why" behind usage data—resulting in little indication of how to design the solution better. ▪ Requires the technical prerequisite that data can be collected. ▪ The focus of customer feedback is mostly on technical problems.

Table 12.8 Profile for the analysis of quantitative usage data method.

12.15 Web Tracking/Click Analysis

Web tracking or *click analysis* is the collection of data using web applications (Hassler, 2012; Beasley, 2013). We have discussed this method in detail in Chapter 9, Section 9.3.8, so we'll keep this discussion brief.

Nevertheless, we have decided to include this method in this chapter to show that it is suitable for the context of use analysis and the evaluation step. The advantages and disadvantages already presented remain the same when used to evaluate design solutions. We, therefore, recommend its use primarily in combination with qualitative methods, such as a usability and UX test.

Table 12.9 shows the profile for this method.

Profile: Web Tracking/Click Analysis	
Brief description	Tracks users' movements and actions on websites and in applications.
Advantages	▪ The data reflects actual usage behavior. ▪ No effort is required for additional data collection—neither by you nor by the users. ▪ Large amounts of data are available for evaluation, so many hypotheses can often be tested. ▪ Continuous data collection makes it possible to repeat the evaluation at different points in time.

Table 12.9 Profile for the web tracking/click analysis method.

Profile: Web Tracking/Click Analysis	
Disadvantages	■ This method only works if there is already a web application. ■ Poor understanding of the "why" behind usage data—resulting in little indication of how to design the solution better. ■ Requires the technical prerequisite that data can be collected.

Table 12.9 Profile for the web tracking/click analysis method. (Cont.)

12.16 Diary Study

In a *diary study*, subjects use an already completed system, product, or service and are asked to evaluate it regularly. Usually, a diary study is performed once a day over a more extended period, generally at least 7 days, but more often, 14 days (Ohly et al., 2010; Döring & Bortz, 2016).

12.16.1 How Does It Work?

Conducting a diary study usually starts with selecting the right participants. People who fulfill specific criteria are typically selected to obtain good results in a diary study. On the one hand, they belong to the target group, and on the other hand, they (would) use the system, product, or service regularly or even frequently. Figure 12.19 shows the schematic sequence for this method.

We recommend you look for more participants for a diary study than you would like to interview. Experience shows that nowhere near all people complete the study. However, how many people ultimately drop out depends on the survey duration, the diary size that needs to be completed, and the *incentive* that participants receive. As you can imagine, the longer the duration, the more dropouts there are. The more elaborate the study, the more dropouts you have. The more money is promised, the fewer dropouts you have. Strive for a good balance for your use case. In our case, typical dropout rates are between 10% and 30%.

Subsequently, *onboarding* of the participants takes place. The exact procedure of this onboarding depends on the objective of the study. Sometimes, onboarding is exclusively a *written preliminary interview* of the participants, in which the persons' previous experiences are recorded through a questionnaire. In most cases, however, an *initial interview* is conducted in which participants are asked in detail about their experience with alternative systems, products, or services; the problem situation; and their current solutions. Depending on the complexity of your solution, an *introduction to your solution* may also be part of the study. If so, the introduction should occur after the initial interview. However, most studies have no solution introduction, as learning and getting to know the system, product, or service is also part of the study.

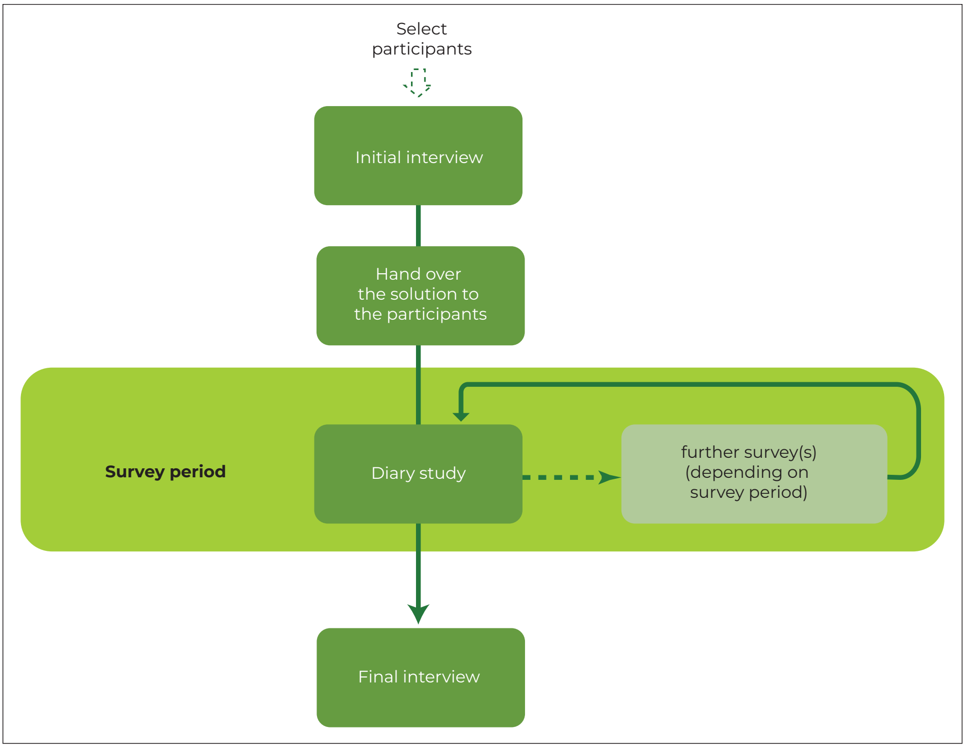

Select participants
Initial interview
Hand over the solution to the participants
Survey period
Diary study
further survey(s) (depending on survey period)
Final interview

Figure 12.19 Schematic representation of the process of a diary study.

After the initial interview, regular use of your solution begins. For this method to work, an essential requirement is that the study only takes place with the finished solution or a close-to-production prototype. Otherwise, problems or—depending on the application—even injuries may occur in the study. Although the initial survey may take place in an official setting, the charm of diary studies is that the system, product, or service is usually handed over to the participants for use in ordinary everyday life. Thus, the users do not come in somewhere to test but simply use the solution as it fits into their everyday lives. In general, however, there are two variants for study planning, which we naturally do not want to deprive you of at this point:

- Participants will be asked to use the system, product, or service daily for the study.
- Participants are encouraged to use the system, product, or service as usual.

You should choose the first variant if you want to understand how quickly the use of the solution can be learned and what problems are associated with frequent use. The second variant is beneficial if you want to learn more about the actual use of the system, product, or service in everyday life.

During the entire period of use, participants are asked to submit a daily evaluation, usually, involving questions about the following topics:

- Frequency of use
- Type of use
- Learnability
- Usability
- Usefulness
- Overall evaluation
- Recommendations
- Special positive or negative events in connection with the product

The trick in designing a diary study is to create a set of questions that give you insight into the use and evaluation of the solution, while also taking very little time to complete. Otherwise, especially with longer periods, participants will not likely use the diary. For our studies, a maximum duration of 10 minutes per day has been found to be acceptable—and if this time commitment can be shorter, all the better!

The questions from the diary can be provided to the participants in different ways. In the past, diaries or paper questionnaires were used for this purpose. Now, the survey is almost universally conducted digitally using questionnaire tools. To increase the likelihood that participants will complete the questionnaire, we usually choose a format where the questionnaire can be completed on a cell phone. We also send reminder emails or text messages to anyone who has not completed the tasks by a particular time. For some studies, however, participants are also asked to keep a photo diary to record their usage of pictures. The smartphone often lends itself to this format as well.

In the case of a survey longer than two weeks, a *more detailed survey* often occurs in the middle of the survey period, perhaps another interview or a more detailed version of the diary questionnaire. Whether you should use a questionnaire or an interview depends on whether you have enough human resources to conduct more interviews. On the other hand, this choice also depends on your research question. Whenever you're still open and want to learn a lot, we recommend you go in the direction of an interview. If your goal is more about reporting or generating "marketing numbers," then written interviews using questionnaires may also be suitable.

In addition, a longer interview always takes place at the end of the survey period. This final interview is conducted in person, remotely, or via telephone interview. We do not know of any diary study that uses only questionnaires at the end, which would lose a lot of important information.

After the end of the diary study, the data is evaluated. In this step, emphasis is placed on various aspects of the evaluation. The most important measures usually include the following:

- Initial assessment of the system, product, or service
- Change of the rating in the course

- Drastic changes in the valuation
- Drastic experiences with the solution

A well-constructed diary study brings a wealth of information that allows quantitative and qualitative data to be linked. In particular, bringing together information about decisive experiences and changes in evaluation is helpful for understanding what is already going well or poorly in the solution.

The results of the study can then be presented in different ways. Typically, a progression over time is presented, with the more significant survey time points highlighted separately. One of the biggest challenges in diary studies is getting participants to participate in the complete study. To accomplish this task, participants regularly receive relatively large compensations. Additionally, compensation for participation is not paid all at once but is split over the life of the study, depending on participation. We usually divide compensation so that an initial share is given after the first interview, then after the interim interview, and the largest share is paid after the final interview to motivate participants to stay until the end.

12.16.2 When Is It Used?

Diary studies are used primarily at two points in time:

- *In the context of use analysis* to understand the current state. This understanding can be about the problem and using predecessor or competitor products to get good input for the later design.
- *In testing advanced product prototypes or even finished as well as new products.* The aim in this context is to understand how the perception of a product changes throughout use and to eliminate potential stumbling blocks in the product that stand in the way of long-term adaptation before the proper market launch.

We see the focus more on evaluation than the use context analysis because often too little qualitative input can be gained to build a basic understanding. Therefore, we'll focus on the use of the method in this section.

12.16.3 What Do You Need?

You do not need much for a diary study. The core of the diary study is usually the survey guide, generally created in a long version and a short version. The short version is used for the daily interviews, and the long version is used for special times, like at the beginning of the study, in the middle, or at the end. The other materials for a diary study are limited to the following items:

- A way to identify and select suitable participants
- Diaries or digital tools for the survey

- The short version and the long version of the survey guide
- A tool for data analysis

A word about the *survey period*: We already mentioned at the beginning that this should be at least seven days. The exact length of the survey period should depend on the system's complexity, the product or the service, and the frequency of use. As a rule of thumb, the more complex the product and the less frequently it is used, the longer the survey period should be.

12.16.4 What Are the Advantages?

The diary study involves monitoring usage over a more extended period in the subjects' typical usage environment. This method also allows effects to be made visible that cannot otherwise be observed in typical usability and UX tests. This aspect applies above all to changes in assessment over time and social influences due to the presence of other people.

Another advantage is that the participants get to know the system, the product, or the service much better than would be possible in half an hour or even two hours of usability and UX testing. This tactic filters out what is essential or obstructive for the users in everyday life, and more minor aspects recede into the background.

12.16.5 What Are the Disadvantages?

A major disadvantage of diary studies is the relatively high dropout rate. The effect of diary studies can be compared somewhat to New Year's Day resolutions: Committing to a fitness program seemed like a good idea then, but after two weeks, not everyone with a gym membership is still committed. The same is true with diary studies.

To prevent the loss of participants, the organization and selection of participants are much more critical and require more time than comparable one-off studies. It is much easier to take half a day off once than to remember to invest 10 minutes every day.

To reduce the number of dropouts, most diaries are kept short and are mainly limited to scales and closed questions, which severely limits the amount of information available in diary studies. Alternatively, you can personally remind participants to participate and complete their tasks, which also reduces the dropout rate, but also increases your effort.

12.16.6 What Are the Alternatives?

Suppose your product is a web application or an app. In this case, you can also use web tracking (see Chapter 9, Section 9.3.8, and Section 12.15) to collect user behavior information over time. The same suggestion applies to all other applications that allow you to track quantitative usage data.

However, a diary study is the only method that allows a subjective assessment of the UX at many different points in time. Thus, the data is not comparable.

Alternatively, you can dispense with the intermediate surveys and only conduct interviews or usability and UX tests. However, this approach will always leave you with a contrived atmosphere, and you may miss a lot of data points. Thus, if you're interested in an authentic evaluation in an actual usage context over a longer period of time, no more meaningful alternative is available than a diary study.

Table 12.10 shows the profile for this method.

Profile: Diary Study	
Brief description	A diary study is the daily collection of information created, for example, by filling out questionnaires during the everyday use of the system, product, or service.
Advantages	■ Provides insight into changes to ratings and subjectively perceived usage patterns over time. ■ More realistic testing than in simulated environments due to use in the actual usage environment (e.g., the inclusion of other people). ■ Getting to know the system, product, or service better through extended exposure enables identifying features that will be important in the long term.
Disadvantages	■ High dropout rate due to long duration and little external support. ■ A small amount of data due to short questionnaires. ■ Poorer data quality due to closed or scale questions, which makes interpreting the data difficult.

Table 12.10 Profile for the diary study method.

12.17 Card Sorting

Card sorting is a methodology for developing and testing menu structures with users (Wood & Wood, 2008; Spencer, 2009). We included card sorting among the evaluation methods after lengthy discussion. Card sorting is (at best) implemented to prepare the design, and thus, strictly speaking, it is not a test. Nevertheless, this chapter on evaluation seemed the most appropriate location to cover this topic.

12.17.1 How Does It Work?

To carry out card sorting, you first need the contents of the menu for your product. You can proceed differently, depending on which goal you primarily pursue with card sorting. In general, you can use card sorting to accomplish the following tasks:

- Identify the navigation structure or the menu groups
- Set the names for the individual menu items and groups
- Set both the groups and the designations in one pass

Depending on your objective, you'll prepare the card sorting task according to the following considerations:

- If you mainly want to know which contents belong together from the users' point of view and which do not, the best tactic is to prepare cards with only the menu items as terms.
- If your main goal is to find out what each term should be called, you should prepare cards that do not specify a name but describe the contents of the menu section.
- If you want to do both, you can prepare cards with a term and an explanation. This version of card sorting is a bit more elaborate and involves a simplified version of a usability and UX test.

But one thing at a time: Let's start with the simplest version first.

In the *simplest version*, you perform card sorting by giving users (selected as user group representatives) a stack of cards with terms. The users then have the task of sorting the cards according to how they see the terms fitting together. Again, two variants exist:

1. You have previously determined which menu items should exist and have specified and named the stacks.
2. You're free regarding the menu items and allow the users to distribute the cards in as many piles as they like.

In the first variant, the users only need to decide to which of the groups specified by you a special term belongs. For example, the predefined groups could be "Our products," "About us," "Shop," and "Shopping cart." The card to be sorted could be called "General terms and conditions." The user would then sort the "General terms and conditions" card into one of the four piles, as shown in Figure 12.20.

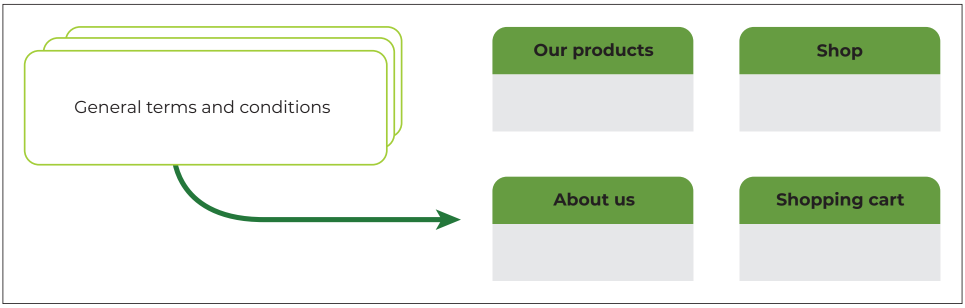


Figure 12.20 Example image for card sorting.

In the *second variant*, neither a fixed number of piles nor names for the piles are given, but the users decide how to sort content. In this variant, the users are still asked at the end to name each pile they have formed.

So, with our example from a moment ago, the user has the “General terms and conditions” card in his hand and no predefined groups. He would put the card in a pile and move on to the next card. In the end, he could call the pile, for example, “Legal,” “Policies,” or “Stuff”—depending on the language usage.

You can continue the naming game as much as you like. If you choose the variant where only the content, but not the term, is written on the individual cards, you can ask for both the stack names and the names of the individual cards after sorting. As you can imagine, this variant is the most elaborate way to perform card sorting—the supreme discipline of card sorting. But you’ll be rewarded with the complete conceptual world of the users.

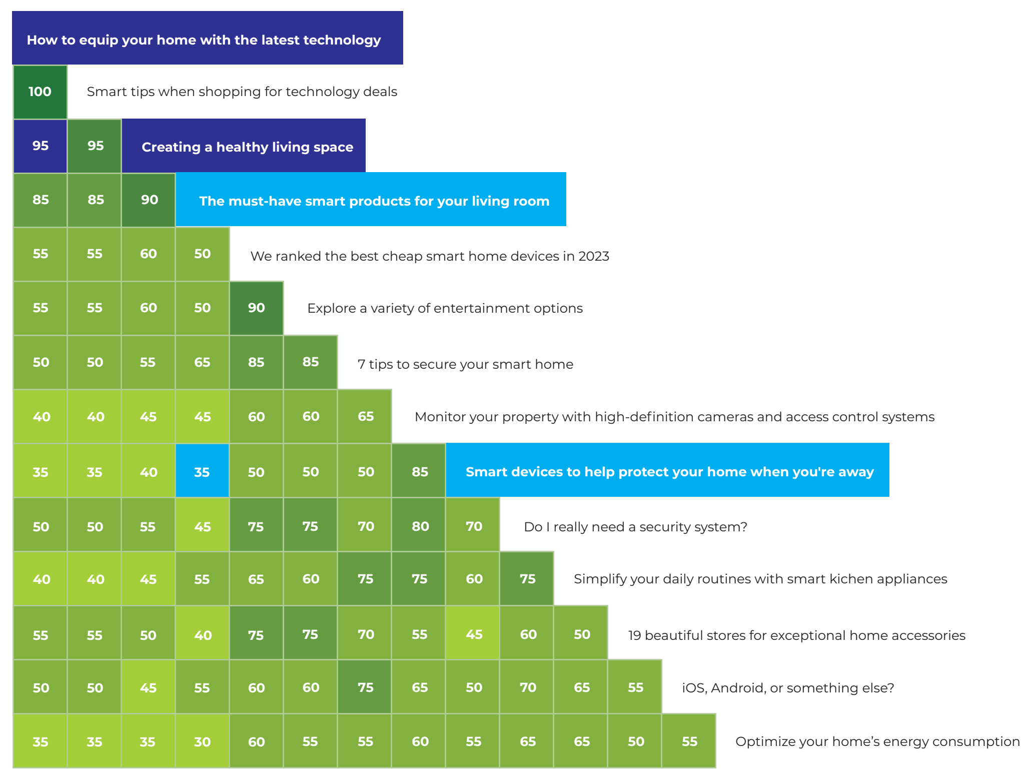
The *middle variant* consists of writing the terms on one side of the card and the explanation on the other. The users are first confronted with the terms and asked to give an accessible explanation. In this way, you learn which terms are self-explanatory and which are not. For all terms where the explanation is wrong, you can turn the card and say, “Actually, we were going for this content. What would you call this?” In this way, you can work your way forward piece by piece but only talk about the areas that are not clear.

No matter how you sort, you’ll end up with different piles and must evaluate the card sorting. Our first card sorting sessions were done with real cards, and the results were calculated in a time-consuming process. Fortunately, this work is now taken off your hands by online tools that can do the evaluation automatically afterward. The number of paid and free tools has increased steadily in recent years, and most of these tools work similarly and allow comparable evaluations. Which tool you prefer in the end is up to you. Our tip: use comparison sites.

The evaluation itself initially refers exclusively to the quantitative part. It calculates how close individual terms are across the various users. Ultimately, you get a statement about which terms you should sort together and which not. You usually don’t receive recommendations but instead get reports that take some getting used to, which show you the similarity of the terms to each other, for example, in the form of a matrix.

Figure 12.21 shows an example that shows that only 35% of respondents would place the terms highlighted in lighter blue in the same category. In contrast, 95% of the respondents would place the terms highlighted in purple in the same category.

Afterward, however, you must decide at which point you want to summarize and where you would still like to keep terms individually. No general rules exist like “Everything from the second level should always be in a group,” but you must decide on each term individually.



													How to equip your home with the latest technology
100													Smart tips when shopping for technology deals
95	95												**Creating a healthy living space**
85	85	90											**The must-have smart products for your living room**
55	55	60	50										We ranked the best cheap smart home devices in 2023
55	55	60	50	90									Explore a variety of entertainment options
50	50	55	65	85	85								7 tips to secure your smart home
40	40	45	45	60	60	65							Monitor your property with high-definition cameras and access control systems
35	35	40	35	50	50	50	85						**Smart devices to help protect your home when you're away**
50	50	55	45	75	75	70	80	70					Do I really need a security system?
40	40	45	55	65	60	75	75	60	75				Simplify your daily routines with smart kichen appliances
55	55	50	40	75	75	70	55	45	60	50			19 beautiful stores for exceptional home accessories
50	50	45	55	60	60	75	65	50	70	65	55		iOS, Android, or something else?
35	35	35	30	60	55	55	60	55	65	65	50	55	Optimize your home’s energy consumption

Figure 12.21 Evaluation of card sorting using a similarity matrix.

12.17.2 When Is It Used?

Card sorting is best used when you’re in the early stages of redeveloping a menu structure and want to determine what content should be displayed together and what belongs in separate areas.

However, you can also use card sorting in preparation for a redesign if you have an existing solution and want to test it. For this task, you can take the content of an existing menu and have users sort this content.

12.17.3 What Do You Need?

For card sorting to produce the desired result, you should have the desired content in mind and prepare it on (digital) cards. In addition, you’ll need users from the desired user group and a tool for implementing the card sorting task. We advise against manual implementation (using real cards) since the evaluation effort is many times higher.

12.17.4 What Are the Advantages?

By performing card sorting, you can efficiently structure the content of a website or application according to a user's mental model. In this way, you can ensure that most users find content in your menu quickly and easily. In addition, depending on the implementation, you also receive feedback on the naming of the individual elements, which makes it easier for your users to find their way around later.

Since card sorting is a standardized procedure without significant survey components, this method is also one of the few user research methods that can be applied successfully without much methodological knowledge.

12.17.5 What Are the Disadvantages?

Thanks to the fantastic online tools, the effort required to carry out card sorting is relatively manageable. However, depending on the menu size, the preparation of the card sorting session (i.e., the creation of the individual cards) can be quite time-consuming: In one of our first projects, we performed card sorting for the website of the Chamber of Industry and Commerce in Germany, which had more than 2,000 subpages. Even when we omitted a content layer, we still had more than 200 cards. Not only is this a lot of work; it's also not fun for anyone.

Another disadvantage is the lack of clarity in evaluating card sorting. Only a few concrete recommendations exist in this context, so although you can do without a real expert to perform the card sorting, you'll need an expert again to interpret the data and derive the menu structure. Thus, the simplicity of card sorting can be a bit deceptive.

12.17.6 What Are the Alternatives?

We are unaware of any other method for systematically deriving the menu structure with users. For testing the menu structure, you can—in fact, almost always—use usability and UX tests (Section 12.7). This choice will also allow you to learn about possible improvements. Usually, however, you cannot redesign an entire structure systematically with these tests.

In our view, card sorting is integral to improving complex menu structures in which users can switch between many contents independently. However, we would also like to mention that the importance decreases bit by bit the better the search functions are implemented. Particularly in the case of websites, users now often reach the desired subpages directly via search engines.

Table 12.11 shows the profile for this method.

Profile: Card Sorting	
Brief description	Card sorting is a method for developing and testing menu structures with users. In this process, the content of an application is grouped and named from the user's perspective to increase the content's findability.
Advantages	■ Allows structuring the content according to the expectations of users. ■ Straightforward method, which is also well-suited for beginners in user research. ■ This method helps in naming the content so users can easily understand. ■ The effort for the implementation is manageable when using tools.
Disadvantages	■ Preparation of card sorting can be extensive and time-consuming, depending on the content volume. ■ No clear guidelines for evaluating card sorting exist, so there is much room for interpretation.

Table 12.11 Profile for the card sorting method.

12.18 Expert Review

An *expert review* is a group of methods in which the assessment of a system, product, or service is not done by the users themselves but by experts who put themselves in the users' place (Sarodnick & Brau, 2006; Mayer, Schrepp, & Held, 2018).

12.18.1 How Does It Work?

Many different types of expert reviews exist. In this book, we'll focus on the most common approach rather than tacking the multitude of expert review approaches available. We have chosen this focus because we believe that "true" data-driven UX design also needs experts to review and optimize the design. Still, these experts should never be the ones to make the final or sole decision about the design. In this respect, an expert review can only be an intermediate step. Therefore, we don't want to give too much space to expert reviews, but we would still like to present the methodology.

The preparation of an expert review usually starts with providing the solution. This provision should be done in a standardized way. In other words, you should ensure that—if several experts are to evaluate the solution—they all get the same idea of the solution.

After deciding how to present the solution, you must decide what kind of feedback you want. Usually, expert reviews can be conducted with two types of experts:

1. Subject matter experts who are knowledgeable in the application
2. Usability and UX experts who know how to make solutions simple for users

The application experts provide feedback on how well the envisioned solution fits the tasks, problems, and context of use. From the usability and UX experts, you get an assessment of whether the way you're currently planning the solution will be understandable for users.

Depending on your chosen focus, you must attract the appropriate experts. In rare cases, the expertise may overlap, and one application expert may be able to give you a lot of feedback on usability—in such a general way that the feedback applies to several users, not just to this one. Sometimes, the usability and UX expert also happens to be an application expert, but this is relatively rare. When in doubt, invite more experts rather than have too few.

In most expert reviews, the solutions are evaluated by at least two experts. To ensure that the experts can evaluate the application well, present the solution to them initially. This presentation can be done either together or separately. In addition, you'll introduce the tasks and user groups and ask the experts to put themselves in the users' shoes during the review.

Afterward, the evaluation phase usually starts, which can also be done either together or separately. Joint evaluation can often lead to lengthy discussions between the experts, so we recommend an individual evaluation phase followed by a discussion. Various methods exist for the evaluation itself.

The simplest method is a *free evaluation* of the application. For a free evaluation to produce meaningful results, you must ensure the experts are experienced. Even then, however, we recommend that you instead perform a supported evaluation.

In a *supported evaluation*, experts are given guiding questions or heuristics to form a basis for evaluating the application. A *heuristic* is a rule of thumb that helps decision-making when only limited information is available. You can choose your focus for the experts' evaluation if you work with guiding questions.

When working with heuristics, a whole range of criteria has been established that you can use as a basis. An excellent example of criteria can be the seven interaction principles from International Organization for Standardization (ISO) 9241-110:2020, which we presented in detail in Chapter 7, Section 7.4.4. Next, experts review the solution and document where contradictions appear or where design heuristics are violated.

This *documentation* can be performed in different ways. We've had good experiences when—at least with digital applications—the screens are available in a presentation and only need to be commented on. The easiest way to collect this information is to have small areas available on a page besides the screens, which represent the various heuristics in color and text and only need to be copied to the appropriate places on the screens. In this way, you can go through a solution relatively quickly, and at the same time, more detail is provided so that the problem can be seen clearly afterward. Of course, other forms of documentation are also possible; you can do anything from printing to filming and commenting. No matter how you document, you should aim for results that are understandable, even without the experts being present. How long such an evaluation session requires depends heavily on the complexity of the application—from 15 minutes for straightforward solutions to 8 hours or even several days for very complex solutions.

Following the individual assessment, all the experts are usually called back into a room to discuss the results. This meeting ensures that different expertise and focal points are brought together. In addition, the discussion enables a better understanding of the results and, thus, a more significant weighting of the problems.

12.18.2 When Is It Used?

Expert reviews are usually used when the solution idea is still at such an early stage that it becomes difficult for users to imagine how the system, product, or service will later work. However, we now feel that this argument is no longer particularly valid, as prototyping is much faster, easier, and cheaper than even just a few years ago. Thus, in most cases, creating suitable testable prototypes is probably as fast as finding suitable experts.

Another use case for expert reviews is that real users in the field are not always available or could be difficult to obtain. This problem happens especially when internal users of an application are involved and the time for interviews is not provided. Or when the company or client does not want the prototypical design to be shown to users for fear that it will have a negative impact on brand perception (which is often an unfounded fear).

12.18.3 What Do You Need?

To make your expert review work, you need relatively little, although it would be best if you had the following materials on hand:

- The ability to present your solution (via a prototype, concept drawing, or even just a description)
- Application experts and/or usability and UX experts
- Heuristics or questions according to which the experts are to evaluate the design
- A documentation template

12.18.4 What Are the Advantages?

Expert reviews have an advantage in that they can take place at a time when no comprehensive visualization is yet available. This opportunity exists because experts are often already well versed in the application and its context due to their previous experience.

Another advantage of expert reviews is the relatively short lead time required to conduct an expert review. In its simplest form, you can decide in the morning that you want to conduct an expert review and maybe even have the results available by evening—provided that the experts are available.

The final advantage of expert reviews is their manageable price range. You do not need technology or expensive laboratories to conduct an expert review. This low cost is another reason why expert reviews have enjoyed great popularity for years.

12.18.5 What Are the Disadvantages?

Expert reviews have several serious disadvantages that you should be aware of. Probably the biggest disadvantage is that most experts may be able to empathize with the user's processes but bring little or no insight into all the other frameworks and characteristics of the user group. Thus, collecting feedback on the acceptability of a solution that is only evaluated by experts is extremely challenging.

Another disadvantage is that, while expert reviews are relatively inexpensive and require less planning, they are also predominantly single opinions. Collecting feedback from several experts can make costs relatively high, even with an expert review.

12.18.6 What Are the Alternatives?

The information you're trying to gather with an expert review—whether the application fits the tasks and is easy to use—you can discover just as well with usability and UX tests (Section 12.7) or usability and UX sprints (Section 12.8). The effort required to produce a testable prototype is often so low that, in our view, accepting this additional effort and going directly testing a process with users makes sense.

Table 12.12 shows the profile for this method.

Profile: Expert Review	
Brief description	The expert review is representative of an expert's evaluation of systems, products, and services based on guidelines without involving the subsequent users.
Advantages	■ This method can be performed early, as experts can evaluate solutions based on descriptions without an interactive prototype. ■ Little lead time is needed, and little organization required. ■ Fast implementation time. ■ No additional equipment is needed.

Table 12.12 Profile for the expert review method.

Profile: Expert Review	
Disadvantages	■ Not representative of the subsequent evaluation by users since the experts lack insight into the actual usage environment. ■ No realistic assessment of the acceptance of the solution by experts is possible. ■ Either evaluation by a few experts (single opinion) or high costs.

Table 12.12 Profile for the expert review method. (Cont.)

12.19 Association Space Test

We introduced you to the association space in Chapter 7, Section 7.4.1, and in Chapter 11, Section 11.12.2. As a reminder, the *association space* is the set of associations you would like to see associated with your solution. Based on this definition, the ARTTest (from the German **A**ssoziations**r**aum**t**est) is a test procedure that builds on a customer's desired associations and compares the actual associations users have of the product using a questionnaire. The ARTTest is a proprietary implementation for which no literature exists yet, but to our knowledge, the principles of the ARTTest are frequently used in design processes in some comparable form.

12.19.1 How Does It Work?

The execution of an ARTTest is effortless: Users are confronted with the system, product, or service and then must provide an evaluation. The test can be conducted on finished systems, products, or services or on prototypes that give an impression of the solution's visual look and feel.

The confrontation takes place in several stages. First, the initial impression of the system, product, or service is ascertained. Then, participants are asked to think of words that spontaneously come to mind when they first come into contact with the solution. Subsequently, the association space is additionally surveyed in a standardized way. For this purpose, a questionnaire with opposing word pairs is filled out, as shown in Figure 12.22.

After completing the questionnaire, users are given time to *explore* the solution independently. The exploration time depends on the complexity of the solution but is usually somewhere between 5 and 20 minutes. You should determine the exact duration in advance and keep it the same for all participants. After the completion of the exploration phase, a new evaluation takes place. This round starts again with free expression, followed by the evaluation based on the questionnaire.

An average value is subsequently calculated for each contrasting pair across all participants to evaluate the data. This averaging takes place separately for the first assessment and the second assessment. In parallel, the standard deviation is calculated. With this data, you can form a profile for the system, product, or service that represents with which terms the solution is associated.

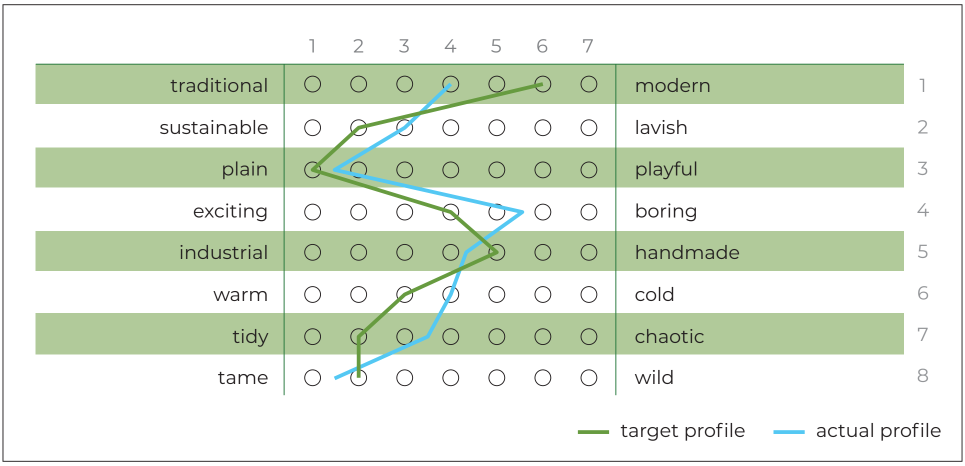


Figure 12.22 Excerpt from the ARTTest questionnaire.

In addition, the free responses of the participants are analyzed. For this purpose, the responses of the various participants are also categorized separately according to primary assessment and secondary assessment, and similar responses are grouped. Subsequently, the categories are ranked, with the category with the most frequent mentions being the *primary association group*, the second most frequent mentions being the *secondary association group,* and so on. Then, you can check whether the categories or synonyms correspond to a word on the list of word pairs. If they do, that term is highlighted as the primary, secondary, or tertiary association. We usually add a term to our word list if no match can be found.

12.19.2 When Is It Used?

The association test can be used for two different purposes:

- To check how the current design affects the users (determination of the status quo)
- For matching a desired association space with an actual association space (determination of the delta)

You can determine the status quo after completing the system, product, or service. However, suppose you want to evoke a particular association with your design specifications. In that case, we recommend that you define the ideal association profile with the stakeholders at the beginning of development. We described how to do this in Chapter 11, Section 11.12.2. Subsequently, you can perform the ARTTest in each development stage with all prototypes that have been visually designed to check whether you achieve the desired association with the design.

12.19.3 What Do You Need?

Not much is needed to perform the ARTTest properly and get the most out of it. The materials and information you'll need include the following:

- A screener to select the appropriate participant group and access the participants
- Visually designed solution or prototype
- Stakeholder-defined target profile of the ARTTest for the solution under test
- ARTTest questions, digitally or on paper

12.19.4 What Are the Advantages?

By using the ARTTest, you can manage the visual design of a system, product, or service in a targeted way. Usually, a lot of effort is invested in defining the brand's core values and thinking about the impression. The development of a design concept also takes a lot of time afterward. In most cases, however, no systematic testing of the design has occurred regarding whether these core values and impressions were achieved with the design. This gap is closed by the ARTTest.

Another advantage is that the ARTTest no longer focuses exclusively on the concept of liking something—unfortunately very subjective—but on whether the desired effect has been achieved. Thus, companies receive a much better depth of information than the conventional way visual design is developed.

12.19.5 What Are the Disadvantages?

Even if the implementation and the ARTTest itself are relatively simple, the definition of the target profile, in particular, requires some work. Sometimes, convincing stakeholders to participate in the necessary workshops is difficult since an analytical approach to measuring visual design is still somewhat unusual.

Another aspect to consider when testing is that great importance must be placed on how the ARTTest is initiated and in what setting it takes place. As soon as the ARTTest is combined with usability and UX tests, you run the risk that the freely stated associations will revolve primarily around usability. Therefore, an essential task to make the initial assessment task-free, relegating tasks preferably to the second survey.

12.19.6 What Are the Alternatives?

Some impressions an ARTTest provides can be captured using standardized procedures such as the AttrakDiff or the UEQ (both described in Section 12.13.1). Especially the scales for hedonic evaluation give a first impression here. However, the use of the procedures is usually limited to measurement; we have not heard of any use for planning so far.

Furthermore, you could also simply collect the associations with free interviews. However, the evaluation effort is incomparably higher, and the degree of standardization is significantly lower. We are unaware of any other method for systematically surveying the association space.

Table 12.13 shows the profile for this method.

Profile: ARTTest	
Brief description	The ARTTest determines which associations the users develop when viewing and using the solution. In this way, the test enables you to test the design quality on a non-functional level.
Advantages	■ Enables targeted control of the visual design. ■ Shifts the focus from subjectively liking the solution to whether the desired associations are triggered.
Disadvantages	■ Requires a definition of the target profile in advance to enable a target/actual comparison—but the definition of the target profile can be pervasive and time-consuming. ■ Requires good methodological knowledge to set the focus of the users' evaluation properly.

Table 12.13 Profile for the ARTTest method.

12.20 Eye Tracking

Eye tracking analyzes eye movements during the usage of a system, product, or service (Nielsen & Pernice, 2010; Holmqvist et al., 2011; Bojko, 2013; Carter & Luke, 2020). This tracking allows not only the actions of the users to be tracked but also allows—at least partially—the tracking of the focus of user attention.

12.20.1 How Does It Work?

Various technical solutions are available for eye tracking, depending on their field of application. Roughly speaking, a distinction can be made between stationary and mobile variants in the following way:

- *Stationary solutions* are always suitable when the person using the device is firmly positioned at a specific location. For example, stationary solution can be used when working on a PC, driving a car (at least to a limited extent), or working at an assembly line or workstation.

  In this case, a stationary eye tracking system is aligned so the person's head is in a headbox (invisible to the user). Within this headbox, users can move relatively freely without the eye tracking system losing their eyes. The headbox itself is usually not very large—often about a 2-foot cube. In return, the tracking does not influence the person using it. Depending on the system, mapping the setup digitally may still be necessary, especially when using stationary systems. Similar to a CAD program, the spatial arrangement of the user and the cameras is displayed so that the system understands what is being measured.

- Whenever the users move more and would thus leave the headbox (or also when the head movements are large), *mobile systems* are a good choice. These systems have an advantage in that they can be taken anywhere. For this purpose, the users wear glasses or a helmet, as shown in Figure 12.23.

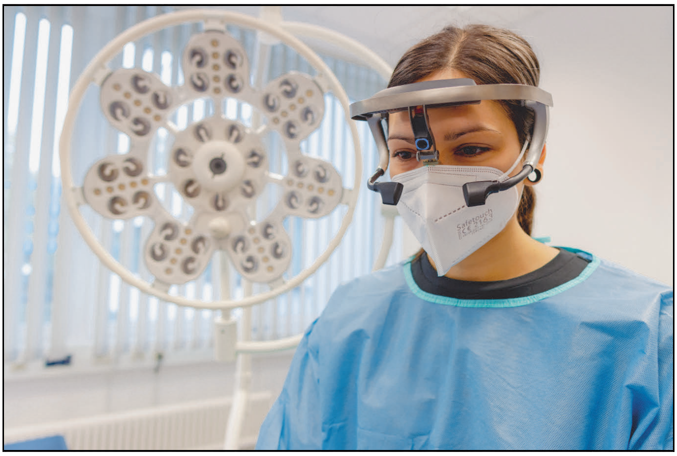
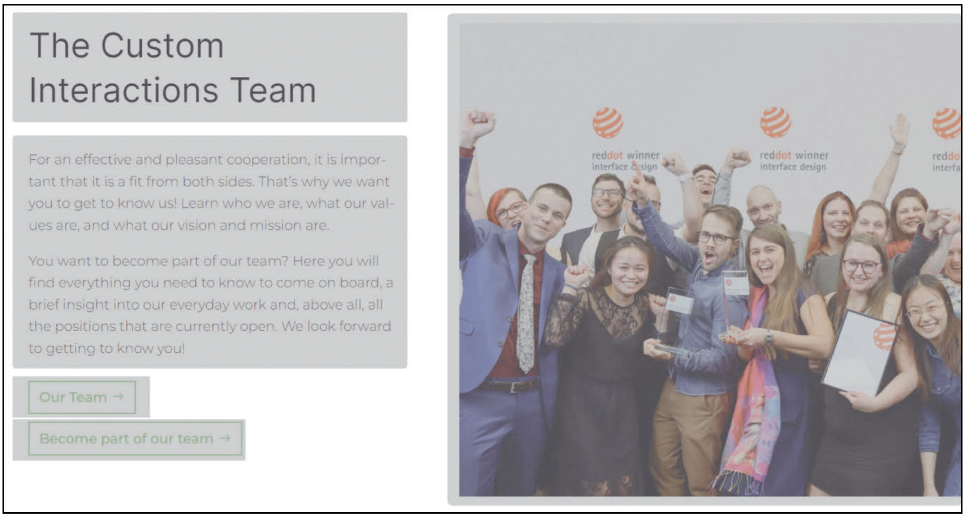


Figure 12.23 Person with head-mounted eye tracking device.

To use eye tracking systems correctly, they must be calibrated for each user before measurement begins. Through calibration, the system learns to detect the user's eye movements and understands when the user looks where. This calibration can vary in complexity and time depending on the system. Without calibration, however, the data is usually not usable.

Eye tracking methods are usually embedded in other test procedures to supplement these procedures. In this case, different data can be generated. In mobile systems, for example, this is a video image in which the eyes' focus is marked as a ring. In addition, data with time stamps on fixation durations and fixation frequencies can be analyzed. However, this information usually requires extensive post-processing in which you set up *areas of interest*. These areas are nothing more than boxes in the image (or in the 3D world); the system counts individually how often and how long a user's gaze falls into the box, as shown in Figure 12.24.

In stationary systems, often used to analyze web pages or computer programs, one of the core results is a *heat map*, as shown in Figure 12.25. In this map, the screen view is overlaid in color, with areas that gathered the most attention usually shown in red. Areas that are not looked at tend not to have any colored overlay. Analysis by fixation frequencies and durations is also possible for mobile systems.



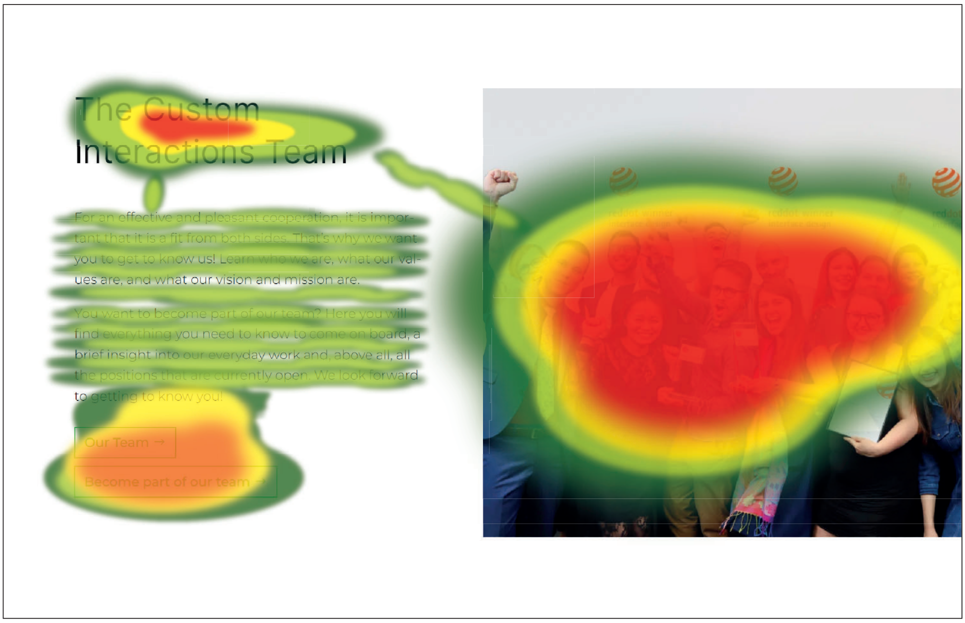
The Custom Interactions Team

For an effective and pleasant cooperation, it is important that it is a fit from both sides. That's why we want you to get to know us! Learn who we are, what our values are, and what our vision and mission are.

You want to become part of our team? Here you will find everything you need to know to come on board, a brief insight into our everyday work and, above all, all the positions that are currently open. We look forward to getting to know you!

Our Team →
Become part of our team →
reddot winner interface design
reddot winner interface design

Figure 12.24 Our website with defined areas of interest.



The Custom Interactions Team

For an effective and pleasant cooperation, it is important that it is a fit from both sides. That's why we want you to get to know us! Learn who we are, what our values are, and what our vision and mission are.

You want to become part of our team? Here you will find everything you need to know to come on board, a brief insight into our everyday work and, above all, all the positions that are currently open. We look forward to getting to know you!

Figure 12.25 Web page overlaid with a heat map to show the distribution of users' gaze in an easily understandable way.

In addition, you can analyze gaze paths for both types of systems. These paths let you know where users' eyes fall first and where their gaze travels. This data can be particularly informative when determining which elements on the web page or product attract the most attention. The distribution of eyeballs helps you figure out which areas attract attention overall. Additionally, you can use eye tracking data to help analyze errors. In particular, eye tracking data helps determine whether an element was not detected (no glances at the element) or not understood correctly (glances at the element but still no usage).

12.20.2 When Is It Used?

Eye tracking can be used at almost any stage of development, from the early prototypes to the finished product. Eye tracking offers particular added value whenever the time is of the essence, for example when discovering a particular piece of information quickly is important, for example, warnings. On the other hand, relevant information may relate to how long the gaze is fixed on a specific element. Eye tracking could be relevant in the automotive context, for example, for checking whether the design of the infotainment system distracts too much from the actual task of driving.

Eye tracking is also frequently used in advertising research, where an advertisement can be tested for whether it attracts enough attention. This testing can be conducted both digitally and in a classic print environment.

We use eye tracking rather sparingly because the added value for the design outside of time-critical applications is quite limited, and a lot of information can also be obtained via other methods (e.g., surveys). Moreover, because the effort to prepare and evaluate the eye tracking study is relatively high, its use is only worthwhile in a few cases.

12.20.3 What Do You Need?

Eye tracking is usually a complementary method for use with other methods. Thus, you'll need everything for the original method, including a testable system, product, or service. The only other thing you need is an eye tracking system. You can usually procure these systems as a complete package, including tracking hardware and evaluation software, so no other components are needed.

12.20.4 What Are the Advantages?

Correctly set up, an eye tracking system provides information about the distribution of attention and the interest of users that would be otherwise difficult or perhaps even impossible to access consciously.

Due to the millisecond analysis of eye movement data, on the one hand, you gain access to unconscious content; on the other hand, you can also prove the security of your systems, products, and services relatively quickly, as you can show quite precisely how long it takes for information to be detected or for the gaze to turn away again. Depending on your field, this information can be crucial evidence that may even come into play in liability issues.

Suppose you use eye tracking system is optimized for stationary use on computer monitors. In that case, you also have a tool that is ready for use and quickly provides interesting data. Depending on the tool, this data collection happens automatically and directly after application use.

12.20.5 What Are the Disadvantages?

Using eye tracking systems usually means significantly more effort for a study setup than without. Almost all common systems (not to say all systems known to us that provide accurate data) require calibration in advance. As a result, a break in the implementation always exists.

When you use a mobile system, users often feel restricted because the systems are bulky, if not uncomfortable, in many cases. The background is that the systems—whether they are glasses or helmets—must fit very tightly and have little movement so that the calibration is not lost.

On the other hand, stationary systems often influence user behavior, as they sit in front of the computer much more stiffly and less dynamically than they usually would, for fear of leaving the headbox.

Another disadvantage of eye tracking systems is the high evaluation effort. Areas of interest must be set manually, especially with a moving image (i.e., primarily with head-mounted systems) where the *area of interest* must be set manually in the evaluation in each video frame. If a user turns his head, the area in the image shifts and must be readjusted. High evaluation effort can also be caused by certain analyses, such as the heat map, as these maps are automated but, unfortunately, only analyze a single person, and the merging of the information for several people must be done manually again.

A rather organizational point that is important, especially compared to other measurement methods, is the price. The acquisition of eye tracking systems is associated with enormous costs.

A final disadvantage already implied in the method's name is the limitation of eye tracking to visually presented content. With the increasing spread of voice interfaces and haptic interfaces, the scope of eye tracking decreases more and more since only very little relevant information for other types of interfaces can be obtained via eye tracking.

12.20.6 What Are the Alternatives?

Few alternatives exist to eye tracking. In PC applications, tracking mouse movements could reasonably indicate a person's attention. However, we no longer consider mouse tracking as a standalone method, as the importance of this method continues to decline. Increasingly, many interactions happen via touch, and thus no mouse is used at all, or many solutions have never had a mouse (e.g., applications in vehicles). Consequently, there is no possibility to collect this data.

If eye tracking is used to perform error analyses, an alternative is a follow-up survey. In many cases, sufficient data about the cause of the error can be obtained by combining observation and questioning.

Table 12.14 shows the profile for this method.

Profile: Eye Tracking	
Brief description	Eye tracking is a method for evaluating systems, products, and services by analyzing users' eye movements. The method focuses on the objective measurement of gaze durations and frequencies and can generate objective statements on whether the users perceived certain design elements.
Advantages	■ Provides objective information about the distribution of attention. ■ This method shows which design elements receive how much attention to a millisecond. ■ Enables the measurement of gaze averaging times. ■ For stationary systems, a relatively low effort for deployment and evaluation.
Disadvantages	■ Most eye tracking systems are expensive. ■ Study design becomes more elaborate because systems usually require calibration—wasting time and interrupting the natural flow of interaction if poorly designed. ■ Users often feel restricted by the systems because they either reduce the range of motion or are uncomfortable to wear. ■ The use of mobile systems results in a high evaluation effort. ■ Only captures visual elements.

Table 12.14 Profile for the eye tracking method.

12.21 Physiological Measurement Methods

Often, demand exists for objective measurement parameters to determine the UX, particularly in the scientific field. Similar to standardized questionnaires, in this section, we won't focus on a single method but instead on a set of methods to objectively measure various physical reactions, which also includes eye tracking. However, since eye tracking plays a much more important role than the other methods, we dedicated a separate section to eye tracking. In contrast, all other methods are considered together in this section.

12.21.1 How Does It Work?

Due to the many possible physiological parameters, listing all the variants of measuring physiological reactions is virtually impossible and also not very useful since using physiological data to determine the UX is extremely rare and has limited significance. Nevertheless, we'll provide a brief overview of what is possible in theory in this section.

As technology advances, you can collect almost any physiological data these days. Whether it makes sense to collect the data depends heavily on your goal.

In an article, Nina Gerber (2016) writes that, while studies aim to establish a connection here, no consistent patterns that apply across the board have yet been found. In her study, Gerber examined the correlation of various physiological metrics with UX, which a focus on the following metrics:

- Electrical activity of the facial muscles as a measure of different emotions (see also Partala, Surakka & Vanhala, 2006)
- Electrodermal activity as a measure of the excitement of a person
- Cardiac activity, measured as heart rate, blood pressure, and heart rate variability, as indicators of valence (liking) as well as excitement
- Respiratory rate as an indicator for predicting emotions and mental stress
- Immunoglobulin A secretion as an indicator of acute stress

All measured values were subsequently compared with various questionnaire responses. Gerber summarizes the rather sobering result of her study with the following words (translated by the authors):

> *"In summary, the measurement of physiological signals currently seems more suited to reveal qualitative differences in the emotional experience of interacting with a product than to capture precise facets of the user experience. Especially for collaborative, social interaction with a product, many confounding factors are present that need to be considered when applying such an approach."*

In other words, no new findings indicate that we could replace a subjective measurement of UX.

All previous measurement methods only provide tendencies or indications of the users' stress levels. However, this measurement is precisely what physiological measurement methods are well suited for. Especially since the advancement of wearables, physiological data can be recorded relatively easily. At this point—without wanting to delve deeper into all measurement methods—we would still recommend collecting physiological data only as an accompanying measure but not as a single measure.

12.21.2 When Is It Used?

Physiological measurement methods (except eye tracking) are mainly used for experimental research. The measurement of physiological data is instrumental when measuring the UX in other ways that would be impossible or dangerous, for example, when driving a vehicle. Data can be obtained continuously based on physiological measurements, whereas a questionnaire can only be completed after the journey.

12.21.3 What Do You Need?

The use of physiological measurement technology is often costly and cumbersome. However, what is needed depends on the respective measuring technique. A detailed list is, therefore, not possible in this section.

12.21.4 What Are the Advantages?

Using physiological measurement data creates a massive amount of data that can be analyzed. As soon as uniform conclusions can be drawn here about the underlying emotion, it enables evaluating products in much greater detail and faster. The actual evaluation always represents an interruption in use: either through an interview or completing a questionnaire. As soon as physiological measurement variables – that are directly related—can be reliably recorded, their occurrence facilitates an evaluation of individual elements.

12.21.5 What Are the Disadvantages?

Collecting physiological data is often expensive and time-consuming outside of wearables, and interpreting the results requires much expertise.

In addition, some measurement methods may require special spatial conditions, such as isolated rooms, to protect the measurement from interference. This makes the methods much less flexible and, therefore, rarely suitable for researching UX outside of a research setting.

Another severe reason against using physiological measurement methods is that the measurement methodology (except for wearables) is often so complex and sometimes even invasive that the UX is significantly influenced.

The last and probably most important point regarding why we would currently advise you against measuring UX based on physiological data is that, at the current point in time, the measures are still far too unspecific and do not allow any conclusions to be drawn about UX—especially since the collection and linking of physiological data and product characteristics have to be done manually.

12.21.6 What Are the Alternatives?

For the acquisition of physiological data, on the one hand, there is the acquisition with wearables, which provide the same kind of data, but are cheaper (and less accurate). If continuous data is to be generated, then eye tracking is also an option (Section 12.20). However, since physiological data is currently very unspecific for capturing the UX, questionnaires or personal interviews are the most suitable means of obtaining information about the UX.

Table 12.15 shows the profile for this method.

Profile: Physiological Measurement Methods	
Brief description	The group of physiological measurement methods is a collective group for all methods that attempt to evaluate a solution's usability and UX based on physiological measured variables (e.g., heart rate variability, pulse, etc.). Since the methods have played a minor role in evaluating systems, products, and services, they are summarized here.
Advantages	■ Most methods produce continuous measurement data so that an evaluation can be carried out event specific. ■ For most physiological measurement procedures, no interruption of use is necessary.
Disadvantages	■ The acquisition of physiological parameters is often expensive (equipment and, if necessary, special spatial conditions) and time-consuming. ■ Evaluation and deployment require a very high level of methodological knowledge. ■ Depending on the method, users are highly influenced by the measurement. ■ Currently, no clear correlations between specific physiological measures and the evaluation of UX have been proven, which is why the evaluation is very interpretative and unreliable.

Table 12.15 Profile of the physiological measurement methods method group.

12.22 Analyzing Errors Using Root Cause Analysis

In the last few pages, we have shown you more or less how to test your product. You learned about many different methods, and we have explained which suits each question. Even though evaluation is the final step in the data-driven UX design process, in most cases, you haven't "made it" yet, but are just now starting the actual work in evaluation. It is a matter of carrying out the evaluations, evaluating them descriptively, and analyzing the results.

As long as you evaluate descriptively, for example, you count errors, measure times, or indicate how many people could successfully solve a task. But what's missing is the question, "Why did this error happen?" Answering this question is the focus of this section.

To determine the root cause of a failure, we recommend performing a *root cause analysis* (Mahto & Kumar, 2008; Wiklund, Dwyer, & Davis, 2016). This method is used whenever you have conducted a test involving the (simulated) product use and observed it.

The basis for the root cause analysis of the errors is the *simplified stimulus-response model of information processing*. We described this model in detail earlier in Chapter 7, Section 7.3. As a reminder: The simplified stimulus-response model of information processing assumes that information processing occurs in four steps and starts with the presence of a stimulus. (So that you don't have to flip through pages, we show the model again in Figure 12.26.)

Subsequently, this stimulus is discovered ("There is something"), then recognized ("This is it"), next a decision is made ("This is what I do"), and this is consequentially executed ("This is how I do it exactly"), which later becomes visible as a reaction. All these steps run one after the other and build upon the previous step. Thus, no stimulus can be recognized that has not been detected before.

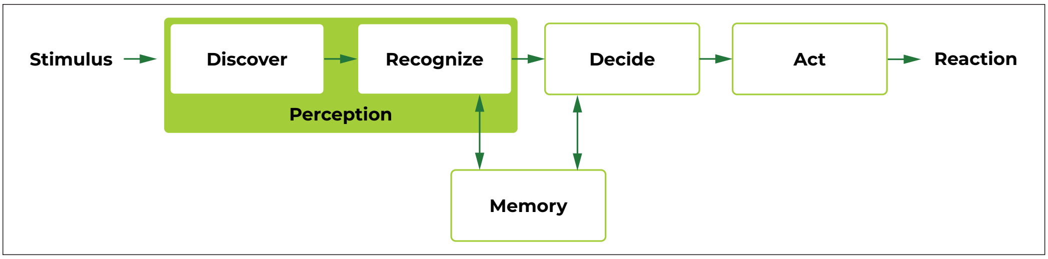


Figure 12.26 The simplified stimulus-response model according to Luczak, 1975.

But how can you now use this knowledge to analyze errors? In principle, it's pretty simple: Errors can happen at any mentioned level, requiring different solutions. Thus, if a stimulus was not *detected*, you need to choose a different design than if a stimulus was not *recognized*. Let's run through this scenario with a concrete example (we'll use the traffic light example from Chapter 7, Section 7.3) to make it easier to understand.

Imagine you're driving a car on the road. The weather is fantastic, and 200 yards away, you can see (detect) a green light. You only need a moment, but you understand it is a traffic light (recognize).

Since the traffic light is green, you decide to continue driving at undiminished speed (decide). Thus, you continue driving straight ahead and do not change your pedal position (act). This example is a good case without any errors, as shown in Figure 12.27. Now, let's look at the same case with errors.

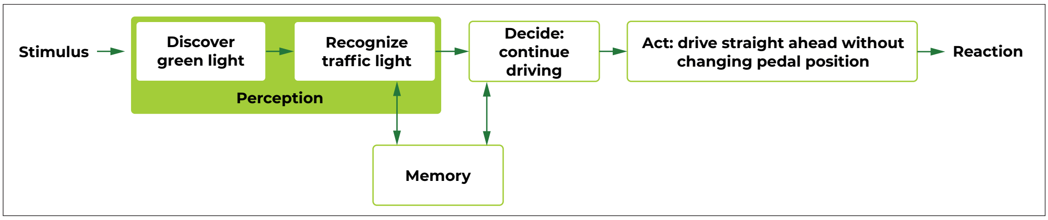

Stimulus
Discover green light
Recognize traffic light
Perception
Decide: continue driving
Act: drive straight ahead without changing pedal position
Reaction
Memory

Figure 12.27 Schematic representation of the simplified stimulus-response model based on the case of "driving a car."

A week later, you're back in the car. But now it is foggy, and visibility is low. You drive the same route, but orientation is much more difficult for you than on the last trip because of the low visibility. You drive more slowly than usual and have difficulty seeing your surroundings. Out of nowhere, another vehicle suddenly appears on your right. Both you and the other driver do an emergency stop, but you can't avoid a minor collision. Fortunately, no one is injured. When you get out to assess the damage, you realize you're in the middle of an intersection. You have entirely missed the red light. In this case, the traffic light detection did not work, as shown in Figure 12.28.

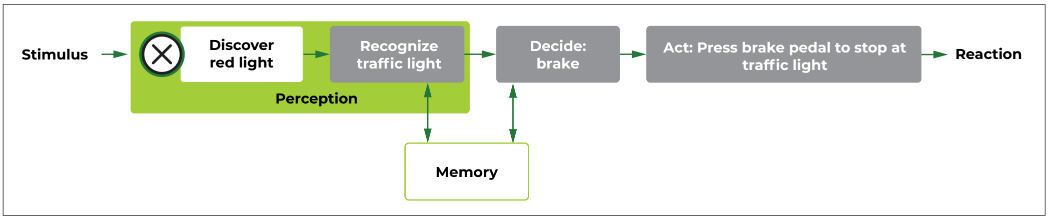

Stimulus
Discover red light
Recognize traffic light
Perception
Decide: brake
Act: Press brake pedal to stop at traffic light
Reaction
Memory

Figure 12.28 Schematic representation of the simplified stimulus-response model with the error at "detection."

We have taken this situation from everyday life to make it easy to understand what a situation can look like where the cause of the problem was a detection error. But it is similar to any other situation where detection is the problem. Usually, you can assume you're encountering a detection problem if you can observe the following behavior:

- The person's gaze does not move or only briefly goes in the direction of the information that should be detected (light, text, display, announcement, etc.).
- The person does not adjust their behavior, or there is no reaction.
- The person does not speak about it unprompted or give any indication of the information.

Simply put, if the important information is ignored, the error is often a "discovery" error. However, you can't be 100% sure until you perform an actual root cause analysis, which includes—in addition to observing the action—questioning the person about their behavior. In this process, a crucial first step is to avoid asking "You made the following mistake; how did it come about?"

We usually start by describing the behavior: "You just kept driving at the same speed for about 15 minutes before you suddenly slammed on the brakes. How did that happen?" In this way, as a first step, you can determine whether the person was aware that an error had occurred. We simply direct the person to the erroneous behavior and ask about the causes. In the case of a discovery error in our example, typical answers might include the following:

- "I didn't even see the traffic light!"
- "It was so foggy; I couldn't detect anything."
- "There was no reason for me to slow down."

All these answers make it clear that the important information was not detected. Importantly, even though the person talks about "detecting" (see the second answer), we can't assume that they describe the actual "detect" step of the simplified stimulus-response model.

Let's imagine the same situation again: You're back in the car; unfortunately, it's foggy again. You're driving more slowly than usual but still have difficulty seeing your surroundings. In front of you in the fog, you faintly see (detect) a yellow light shining. You assume a heavy truck is moving in front of you and continue driving. Out of nowhere, another vehicle suddenly appears on your right. Both you and the other driver make an emergency stop, but you cannot avoid a minor collision. Fortunately, no one is injured. When you get out to assess the damage, you realize you're in the middle of an intersection. You did not realize that the yellow light did not belong to a vehicle but to a traffic light that just changed to red, as shown in Figure 12.29.

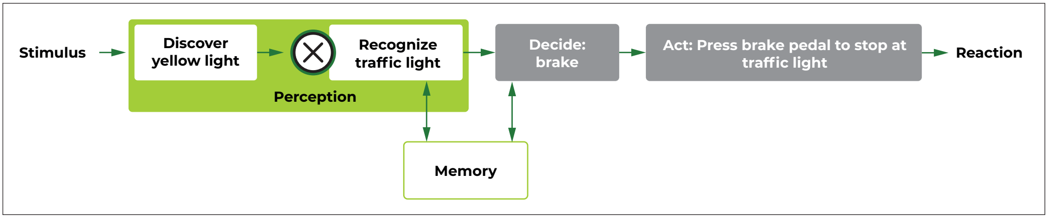


Figure 12.29 Schematic representation of the simplified stimulus-response model with the error at "recognition."

With this example, we hope you can easily see why a root cause analysis usually consists of an observation and a survey. After all, you cannot observe any difference in result between the error in detection and the error in recognition, which is not always the case. Along the way, however, you could have observed subtle changes in behavior, such as the following:

- The person's gaze wanders to the information.
- The person does not adjust their behavior, or there is no expected reaction.
- The person expresses their opinion about the information, if necessary, but in a way that is not appropriate (for example, "Oh no, a heavy truck!").

However, the difference is usually most apparent in the follow-up interview, where you'll probably hear the following sentences:

- "I thought this was a heavy transport."
- "I didn't think it was important because..." (insert an alternative interpretation of the information).

What all these answers have in common is that the respondents were aware of the existence of the information. Still, either they did not realize its importance for the situation, or the information was misinterpreted in terms of content.

Now, imagine you're driving again along the same route. Fortunately, the weather is fantastic. A yellow light suddenly appears in front of you (detect). You immediately realize that it is a traffic light (recognize). You decide you don't want to take risks and would instead stop too early than too late. You, therefore, hit the brakes hard. However, because you were traveling quite slowly due to recent events, you come to a stop about 50 yards before the traffic light. The driver behind you can no longer brake in time and hits your vehicle. Thus, you have discovered everything, recognized everything, and still made an unfavorable decision, as shown in Figure 12.30.

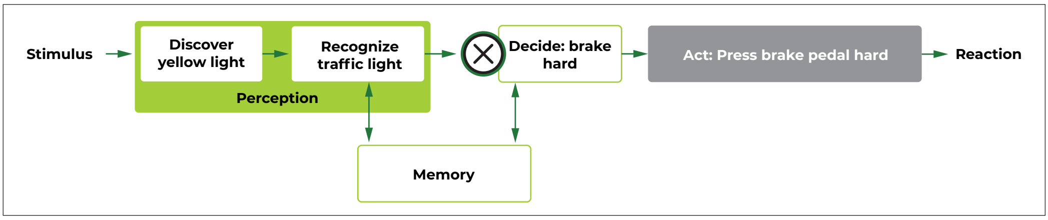


Figure 12.30 Schematic representation of the simplified stimulus-response model with the error in "decision."

Identifying a wrong decision by observation is not so easy. First of all, everything looks the same as it would in the case of a recognition error; the first clues by observing the accompanying statements of the person:

- The person's gaze wanders to the information.
- The person does not adjust their behavior, or there is no expected reaction.
- If necessary, the person comments on the information and shows that they have recognized it. However, the person draws an inappropriate conclusion from it. (Example: "Oh, the traffic light is turning red. Now I have to brake really fast!")

In the conversation that follows, you'll get answers like the following:

- "I really wanted to stop in time."
- "I really thought it would take longer."

So the answers all go in the direction that the person was aware of the relevant information but simply assessed it differently. In the same situation, you as a driver could also have decided that you were in a hurry and therefore accelerated again to avoid waiting at the traffic lights. However, if the traffic light had turned red in the meantime, you would have also made a decision error here.

Especially in the digital world, act errors occur less frequently. For completeness, however, we still want to present them to you.

Imagine again that you're driving in the car. The weather is good; you're in a good mood because there hasn't been an accident in ages. A yellow light appears before you (detect), which you immediately identify (recognize) as a traffic light. You're afraid you won't make it through the intersection when the light turns yellow, so you decide to brake (decide). As you hit the brakes, your shoe slips, and you hit the gas pedal. Your car howls and lurches forward. You ran the red light, but luckily no other car was around. In this case, you detected and recognized all the information correctly. You even made the right decision but still made a mistake in the action that almost led to a tragic outcome, as shown in Figure 12.31.

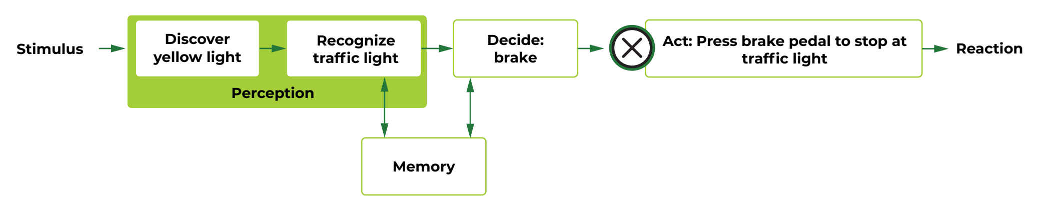


Figure 12.31 Schematic representation of the simplified stimulus-response model with the error at "act."

Action errors are straightforward to spot:

- The person attempts to perform an action but cannot complete it successfully or only partially.
- The person performs the wrong action (e.g., pressing a slider button).

There are very few ways to miss an action error. The action is by far the most obvious part of the simplified stimulus-response model. Nevertheless, don't forget follow-up questions, which can provide more clues about causality and certainty. For example, you might hear the following statements:

- “The pedal was tough to push; I just slipped off.”
- “The sole of my shoe was extremely slippery today, so I slipped.”
- “It was raining outside, and the pedal was wet, so I slipped.”

The result—namely, the slippage—was the same for all answers. However, the cause of the slippage would have been explained differently in all cases.

In this way, you can review each product error and have an overview of the suspected causes. Our recommendation: Don’t rely solely on assessments by the participants, but also add your expert know-how. Once that’s done, then it’s on to the revision.

12.23 Learning from the Data and Making Informed Decisions

Suppose you have taken our recommendation from the last section seriously. In that case, you should now have a great understanding of the causes of the errors and problems you observed in your usability and UX tests. Furthermore, suppose you have defined a desired profile for the perception of your application in advance (for example, via the ARTTest). In that case, you should understand how close you are to the users’ desired evaluation of your solution.

12.23.1 Implementing Feedback

When deciding how to deal with this feedback, you should first make a very clear distinction between *objective feedback* and *subjective feedback*:

- All *objective data* about how well the solution works (i.e., errors, times, detours, comprehension) should be used to decide whether to change the page layout, screen flow, and the interaction itself. If no objective problems exist, you don’t necessarily have to change anything.
- The *subjective feedback* on whether your solution triggers the desired associations is perceived as pleasant and fitting to one’s tasks, predominantly providing you with information on whether and what should still be adjusted in the visual design.

Of course, both areas are related, and a fundamental change in the visual design can also lead to adjustments in the interaction. However, as a first step, you should consider them separately.

A helpful approach is to work with variants and incremental tweaks to make visual adjustments that let the user perceive that their associations are aligned with the application. Working with profiles and pairs of opposites, in particular, will give you clues about adjusting the design. For example, if you want to be perceived as natural, but the perception is more technical, then a step in the right direction might be to adjust the shapes, fonts, and colors, moving toward rounder elements and more natural colors, as shown in Figure 12.32.

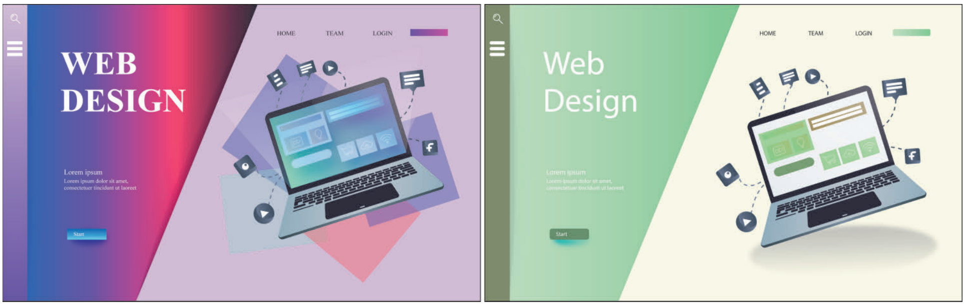


Figure 12.32 Example optimization of an application towards the desired association "natural."

However, we have observed among our customers that many find it significantly more difficult to meaningfully eliminate the causes of usability problems. You have already taken the first step by analyzing the causes. The second step is to derive solutions. This step is relatively simple because you have already obtained good indications of a possible solution by classifying the causes of errors according to the simplified stimulus-response information processing model.

The basic scheme for fixing problems at each level can be described in the following way:

- **Detection problem**
  Emphasize information by placement (higher up, more to the left in Western culture); size (larger); color (flashier, more noticeable); and shape (different from the others).
- **Recognition problem**
  Revise text or icon choice; choose less abstract representations.
- **Decision problem**
  Provide decision support such as activity instructions, alarm limits, and recommendations.
- **Action problem**
  Optimize the physical attributes of the respective controls (size, grip, distance, shape, force required).

Thus, if you know the cause of a usability problem, all you have to do afterward is figure out which revision options are the best. Often, however, this choice becomes clear when you look closely at the error and check what was actually said in the tests. If not the case, you can still iterate further and test again—which is what we recommend in any case.

12.23.2 Documenting Decisions

We want to address the topic of documentation briefly. We know documentation is an unpopular topic, and nobody wants to document anything. However, we have noticed that, especially in projects that are accompanied by many stakeholders or that run over a long time, there is a tendency to develop a design, test it, adapt it, and then fall back into the old ideas after a while. The result is a lack of long-term learning, only short-term learning.

Of course, the best way to prevent this problem is to have stringent logic according to which the usability and UX design was developed and implemented. We've tried to explain this logic in the last 600 pages. Now, the only remaining task is to make the logic visible. We do not want to give you a form to fill out. The only rule we suggest is this: *Do not only document the results but also the reason for the decisions!*

12.24 Evaluations in Medical Device Development

At the very beginning of this chapter, we discussed the topic of formative versus summative evaluations (Section 12.1). We listed the differences between a test that is performed during development with the aim of improvement and a test that takes place at the end of development with the aim of proving quality. We would like to briefly discuss a special case, namely, proving quality at the end of developing a medical device.

Suppose you manufacture a medical device or an in vitro diagnostic device. In that case, you're required by the FDA in the US or in the EU the MDR (*Medical Device Regulation*) or IVDR (*In Vitro Diagnostic Regulation*) to provide evidence of the device's essential safety and performance requirements. This process includes—among many other things—proving that you have done good usability engineering for your product and that these efforts have been successful. You can prove the success of your usability engineering activities by carrying out a summative usability study—before your product is used by patients or users for the first time.

In this context, too, "summative" means that you assess the quality of the product at the end of development. You're also no longer allowed to work with prototypes but must test with the pilot series, the finished system, or the product itself. This barrier is the only way to ensure the results apply to the product. If you're in a medical context, relatively straightforward but strict requirements for usability engineering and the associated study must be followed. In addition to the many little things you must consider in this context, we would like to point out three important considerations, where many mistakes happen:

1. You'll need formal proof of your qualifications to work in the medical industry as a usability engineer.

2. With the summative study of a medical device or in vitro diagnostic device, you prove that it can be used without posing an unacceptable risk to users, patients, and third parties. This study is, therefore, purely about safety and not about UX.
3. Strict rules for interacting with participants exist during the summative study, which differs significantly from how other tests are administered.

Our goal in this discussion is not to provide instructions on what a summative study in medical technology must look like. Instead, we want to raise awareness among you that—although everything is called "usability testing"—the rules can differ in various fields. Don't be surprised if you need to understand "summative study" entirely differently (in this context) than in all fields outside of medical devices.

12.25 Example: Evaluation Methods for moveHUB

Matthew, Emma, and Anthony are nervous. The first small user test for the mobility app is about to begin. For this purpose, they built a simple wireframe click dummy, as shown in Figure 12.33, and invited some student friends. They know a test with friends is not ideal, but they want to start somewhere first. In addition to the click dummy, they have prepared a test guide. Their goal is to use this usability and UX test to understand if the general flow in the application works for the use case "suggest alternatives." Do users understand that they will be late if they stay in the current mode of transport? Do users understand the alternatives? Can they switch to them?

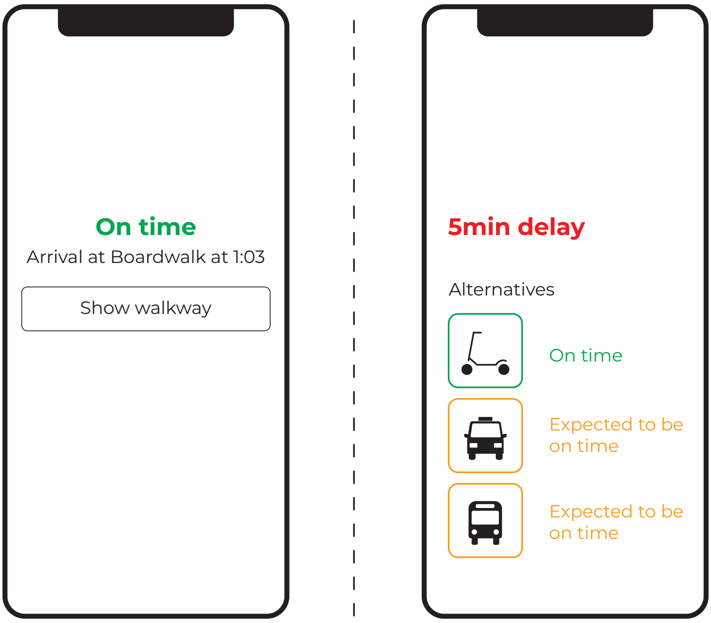


Figure 12.33 Wireframe click dummy for moveHUB: No delay is expected (left). The app suggests alternatives due to the delay (right).

Since the prototype is only running on a computer so far, the three have decided that Matthew will conduct the interviews in a quieter café area in the cafeteria. Matthew's computer is in a web call, where Emma and Anthony are also located. The two are also sitting in the café area, but further away at a different table. This setup allows everyone to follow and record the usability and UX test live without disturbing the user with many direct observers.

After the tests, the three draw first their feedback. They want to take a closer look at the individual data, but what has already become apparent is that the basic process is understood. Various points where they can still improve stand out: In particular, all users raised the question of whether they have to look at the app actively all the time or whether it works in the background and notifies them when they have to change their means of transport. Acceptance also seems to be linked to this capability. Most respondents only want to use the app when they need an alternative and don't need to check it all the time. Another point often mentioned is the cost. Users want to be able to weigh delays against costs and decide whether to continue their journey at all.

In general, the three realize that costs are an essential issue. Another central insight arose, even if it wasn't the focus: They learned through their first contact with users that the app shouldn't cost anything, certainly not monthly and that, ideally, it shouldn't contain any advertising. This last point shatters the three founders' funding idea. While they haven't invested much time in it yet, they were previously thinking of a monthly payment model of $1 under the slogan "Your dollar for punctuality." But that seems out of the question since the app costs are supplemented by the travel costs of e-scooters, for example.

On the whole, the three students are satisfied. They have learned several new things that must be sorted out and incorporated. Therefore, they decide to delve deeply into the evaluation and tackle the next big problem. Their idea: If we first take care of the *showstoppers* in all areas—in other words, the points that would lead to failure—then nothing will stand in the way of our success.

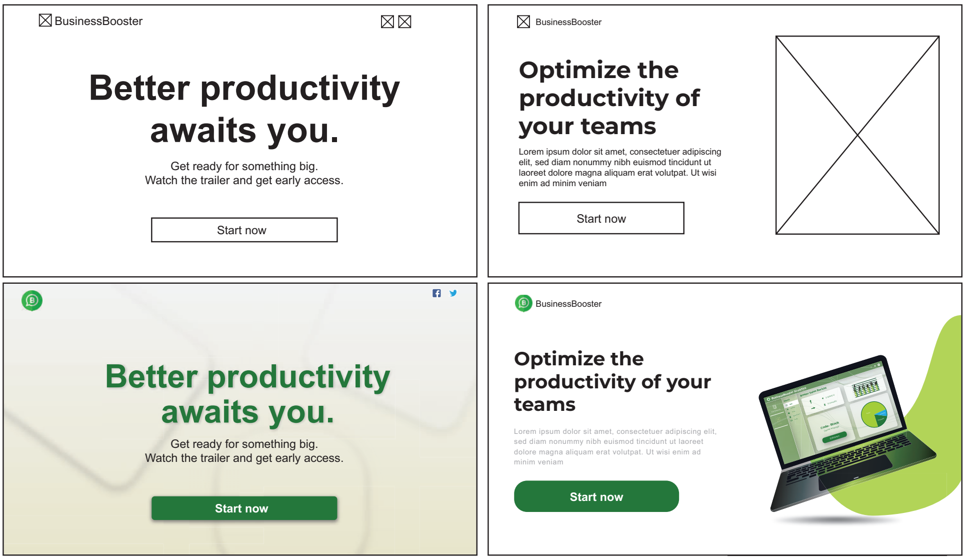
12.26 Example: Evaluation Methods for BusinessBooster

We meet BusinessBooster again at a time when some user tests have already run. The tests were conducted as usability and UX tests by an external data-driven UX design agency. Markus Meier wanted his team to focus on development, which seems to be working out. The developers didn't have to worry much about the test scenarios and test execution, and they gathered a great deal of new knowledge that they were able to integrate step by step.

Today, we have a special usability and UX test coming up. Remember, BusinessBooster sells by providing a demo of the software for free. After demo use, then prospects either become customers or stop using it. Therefore, making prospects aware of BusinessBooster and the demo in the first place is an important step. The upcoming usability and UX test revolves around this moment and is particularly exciting for the team to observe.

At first, Markus and his team didn't think about this part at all. However, the external data-driven UX design agency refocused the strategy: Not only designing the product itself would be beneficial but also designing the sale (and thereby facilitating turning potential prospects into customers). This endeavor immediately made sense to Markus: Many times before, a situation during a trade fair had come up when he spoke with representatives of companies that had just decided on a new solution without even knowing BusinessBooster.

Therefore, a landing page test is scheduled for this test with the following idea: First, BusinessBooster is advertised via various measures, such as search engine advertising and flyers at trade fairs. Thus, interested parties are always directed to this website where the product and its benefits are explained. Furthermore, the free demo is shown on this website, and ideally, it is downloaded by interested parties. Moreover, a kind of "registration" also takes place so that the sales department of BusinessBooster can actively contact. Different variants were designed with the agency, which is to be compared now with the advertisement—still in the wireframe condition. In addition, initial opinions on a possible visual design will also be obtained, as shown in Figure 12.34.



Better productivity awaits you.

Get ready for something big.
Watch the trailer and get early access.

Optimize the productivity of your teams

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam

Figure 12.34 Two variants of the BusinessBooster landing page. Which version leads to users downloading the demo?

The user tests will initially be conducted with the prototypes as usability and UX tests. The goal is understanding which elements work well on the landing pages and, most importantly, why. The guideline, therefore, includes questions such as the following:

- What keywords would you type into the search engine?
- Have you seen the ad?
- Would you click on the ad?
- What is your first impression of the landing page?
- What information are you missing?
- Would you download the demo?

The interviewer is instructed to ask detailed questions and to determine the underlying "why." The findings should then flow into the revision of the two pages.

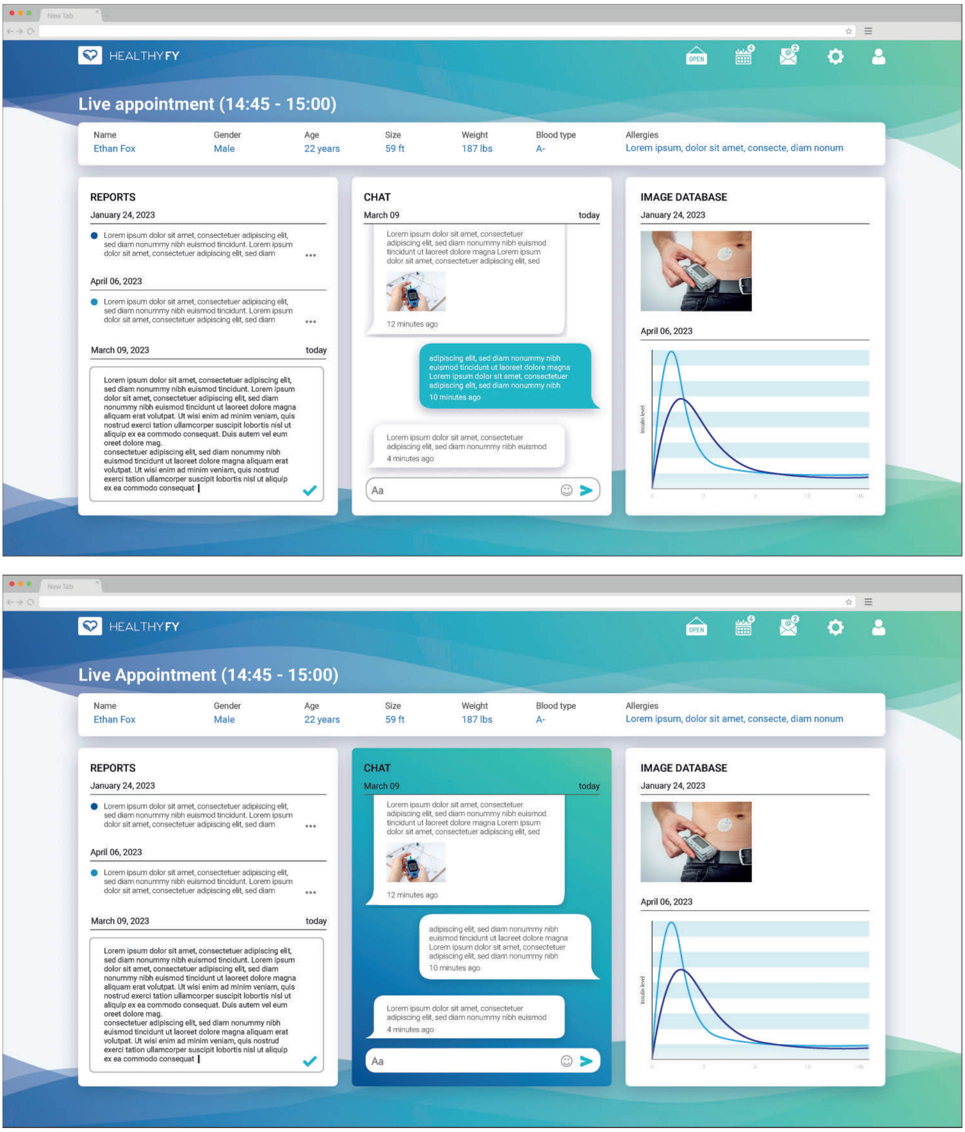
The team then plans to implement the two sites with a web tool and—since Business-Booster already exists in the "old" version—to use it live, including advertising. With this procedure, they intended to supplement the qualitative results from the usability and UX test with quantitative data. The central metrics in this case include the following:

- Number of times the ad is shown
- Number of clicks on the ad
- Display to click ratio of the ad
- Number of visitors on the landing page
- Number of clicks on the demo button
- Number of finished registrations for the demo
- Number of purchases after the demo

Markus is optimistic. He believes that his team and the agency have done an excellent job. Every usability and UX test conducted so far has brought the team a big step forward and measurably improved the prototypes. "Why should it be any different with the landing page?" Markus thinks as he takes his seat in the observation room.

12.27 Example: Evaluation Methods for Healthyfy

We meet Healthyfy again just a few days after our last visit. Most recently, as a result of the discussion in Chapter 11, Section 11.16, the team designed two different variants of the central chat feature to compare different looks, as shown in Figure 12.35.



Live appointment (14:45 - 15:00)

Name: Ethan Fox
Gender: Male
Age: 22 years
Size: 59 ft
Weight: 187 lbs
Blood type: A-
Allergies: Lorem ipsum, dolor sit amet, consecte, diam nonum

REPORTS

January 24, 2023

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam •••

April 06, 2023

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam •••

March 09, 2023 today

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum oreet dolore mag.
consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat ✓

CHAT

March 09 today

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed
12 minutes ago

adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh
10 minutes ago

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
4 minutes ago

Aa

IMAGE DATABASE

January 24, 2023

April 06, 2023

Live Appointment (14:45 - 15:00)

Name: Ethan Fox
Gender: Male
Age: 22 years
Size: 59 ft
Weight: 187 lbs
Blood type: A-
Allergies: Lorem ipsum, dolor sit amet, consecte, diam nonum

REPORTS

January 24, 2023

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam •••

April 06, 2023

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam •••

March 09, 2023 today

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum oreet dolore mag.
consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat ✓

CHAT

March 09 today

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed
12 minutes ago

adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh
10 minutes ago

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
4 minutes ago

Aa

IMAGE DATABASE

January 24, 2023

April 06, 2023

Figure 12.35 Two variants of the central chat feature from the Healthyfy physician view: Which one will users choose?

The team has prepared a quantitative test that first carries out an ARTTest with users and then delves a little more deeply using a few predefined questions. Since the team has already evaluated the general visual design, this test scenario focuses only on the chat area. The following questions are asked:

- Do the above associations still fit the desired ones?
- Which version do users like better?

- Do any of the versions cause usability and UX problems? (We expect a negative answer here because the layout has been tested several times; we're really asking this question again "to be safe.")

The whole test is designed as an online questionnaire to make it accessible to many people. The first part shows the two designs, and users are asked to describe the adjectives freely. The test is designed so that half of the people first see and describe the first variant. Then, they continue with the second variant. For the second half of the participants, the procedure is reversed: first the second variant, and then the first. Thereby, the team tackles sequence effects.

The subsequent question section consists of only four questions:

- Which design do you like better? (The user should choose either one or the other)
- Why do you like the design better? (A free text field)
- Is there anything that would prevent you from using your preferred variant? (A free text field)
- Is there anything else you would like to tell us? (A free text field)

The plan is for at least 30 diabetologists to answer the questionnaire. Afterward, the team would like to start the evaluation and, if necessary, derive measures. The test has been set up, the email addresses of the diabetologists are ready, and now the team has nothing to do but wait for the first responses to roll in. Exciting!

Chapter 13

Design, Get Data, Repeat: When Is It Enough?

"Repeating is harder than anything else."
—Usain Bolt (one of the greatest sprinters of all time)

"When does it actually end?" is a question we've encountered again and again in our professional careers as usability and user experience (UX) designers. Especially when we work with new clients for the first time, the question of planning an iterative process is essential. In this chapter, we would like to share some approaches to help you determine the end of the data-driven UX approach.

First, a disclaimer from our side: *Data-driven UX design is never done.* It's not as if you can "do" usability and UX design once and never have to touch the subject again. On the contrary, once you have decided that you want to have excellent usability and UX for your systems, your products, or your services, then from now on, you'll have a topic that will accompany you for the rest of your product lifecycle.

Every time you release a new version, integrate new features, or plan a design update, you must consider whether you still manage to provide good usability and UX. But even if you don't change anything about your product, your customers' and users' needs and expectations will change over time. These changes alone will mean that, after a while, you'll have to revise the solution.

Thus, usability and UX are a bit like healthy eating and the right weight: It's no use going on a crash diet and eating nothing for a week. With this choice, if everything goes well, you might lose 4 pounds, maybe even 8. If you were overweight before, however, you'll still be hungry and have poor eating habits afterward and will soon gain weight again. To lose weight and stay slim afterward, you must create healthy habits.

Maintaining a good usability and UX in your systems, products, and services is comparable to this problem. Doing usability and UX design once does have an effect and is a good first step. However, this first step can quickly fizzle out. Shifting to a "healthy" habit is needed to keep usability and UX sound in the long run.

Having said that, let's return to the question, "So when does data-driven UX design end?" For us, data-driven UX design involves two phases:

- *The initial phase,* in which the usability and UX concept of the application is defined
- *The second, continuous phase*, in which care is taken to ensure that the concept is also implemented and further developed

The initial first phase has an end and moves into the second phase. The second phase is ongoing as long as the system, product, or service continues to be developed. We, therefore, want to focus on the question of when the moment has come to move from phase 1 to phase 2.

You can define the transition moment in different ways. From our point of view, there are two main approaches you can take:

- You can determine the transition based on the quality.
- You can determine the transition based on the time.

What exactly do the two variants mean, and when do we recommend which variant? Let's now answer these questions together.

13.1 Determining the Transition Based on Quality

Recall Chapter 3, where we initially discussed the concept of return on investment (ROI) in usability and UX design. One of our considerations was to think about what you want to achieve by setting up and applying a data-driven UX design approach. Now is when you should pull these considerations out of your bag again.

If you're following a data-driven UX design approach (or any other form of human-centered design approach), you're most likely taking an iterative approach. Nevertheless, from our point of view, each iteration represents a kind of closed cycle in which the individual steps of the data-driven UX design process are run through once. Thus, when you get to the final stage, evaluation, and look at your results, it's time to consider whether another loop is needed. To make this consideration not purely based on your gut feelings but instead on resilient and meaningful data, we recommend referring to the *key performance indicators (KPIs)* that you initially defined for the desired ROI.

Before you start testing, you'll need these KPIs to define what you should test in the first place (we went into this in detail in Chapter 12) and afterward, when you need to interpret the results.

The principle is quite simple: If the characteristics of the KPIs that appear in the test correspond to the characteristics that you planned and would like to achieve, then you're done testing at this point. Instead of planning another iteration, you look at the results of the last test again in detail and consider which of the discovered opportunities for improvement you still want to implement.

If you have achieved all the KPIs as you wanted to achieve them before, then it can also be an excellent decision point to say, “We’re not implementing anything anymore; we’re done!” The only exception is when you have collected feedback that makes use impossible and paralyzes the whole system, product, or service.

An example of this scenario in the video game industry is called a *game-breaking bug*, that is, a bug that causes the entire game to stop working. For example, if a non-player character (NPC; a character played by the computer) is central to the main story but is missing due to a bug, the player cannot talk to the NPC, and ultimately, the game cannot be continued.

Of course, you still have to fix this error even if everything else fits. But otherwise, your measured values represent what you defined beforehand: your *quality gate*, that is, the threshold that your system, product, or service must pass so that you say, “Now it’s good enough.” So don’t worry if you still find one or the other more minor issues at this point. It’s always going to be that way.

However, if you’re achieving your KPIs precisely as you’d like but notice from user reactions, “That’s not it yet!” you can always readjust your quality goals and redefine what you want to achieve before you start implementing or going live.

Be aware of falling into the typical “technician trap.” Technicians, engineers, and many other people in technical professions are trained to do everything a hundred percent perfectly. This approach is good and necessary in many areas. However, when you apply this logic to an entirely new product, you don’t want to spend years tinkering with features while your competitors are already busy winning customers. You should only take this risk if you’re sure that the features will bring the desired added value—according to the opinion of your customers and users, not your opinion.

13.2 Determining the Transition Based on Time

The alternative to the quality approach is the *time approach*. Quality is essential for success, but time is often the deciding factor between victory and defeat, especially in a startup. This relationship exists mainly because most startups aren’t making money yet, thus making it easy to calculate the time it takes to finish the startup: Simply look at the available capital and then divide it by the monthly costs.

In such a case, we avoid putting the cart before the horse by following these steps:

1. Think about how long your money will last.
2. Next, consider how long it will likely take to generate enough paying customers (or get funding) to support your costs.
3. Then, consider how long it will take you to implement it.

4. And now, if you take a timeline or a calendar and plot the times backward from the "money running out" date, you'll know how much time you have left for a data-driven UX design approach.

A little tip from us: Always plan a buffer. Experience shows that everything takes twice as long and is twice as expensive.

In our experience, no minimum amount of time is required to take a data-driven UX approach. Instead, we have a minimum number of iterations you should go through. The pace at which you go through these iterations depends entirely on your planning and the amount of time you can use.

Depending on the customer, we know a pretty wide variation in this context: We have customers who are relatively relaxed because they want to coordinate everything internally with many people. An iteration may take one or sometimes even two months. Then, we have customers who want to finish as quickly as possible. With these customers, we iterate daily, including testing and design. But of course, the daily iterations only work if you have good and easy access to the group of users.

On the one hand, you can check how much time you could invest maximally before your company runs into economic difficulties. In this case, you're then allowed to be faster at any time. On the other hand, you can consider how much time you want to take. Of course, this leisure only works if you're funded; otherwise, you may get a nasty surprise. We want to give you the following empirical values to help you decide how much time you want to invest: We usually need two to three iterations to design two to three core workflows with one target group. The more workflows and different target groups there are, the more iterations you'll usually need.

Based on your company, you should plan how long each iteration should take. If you have been in your company for a while, then you can estimate how long you need for feedback, internal consultations, or even planning. If you can't do that (yet), then draw upon your colleagues or contacts. Especially if there is little experience with the agile approach and data-driven UX design, you should plan more time than is assumed from the start. If you're then faster, this is usually not bad. However, we often experience that times are planned significantly shorter than needed. This miscalculation causes delays and frustration but, above all, a negative mood. You can avoid this negativity if you remain realistic right from the start.

13.3 Should You Readjust? The Flexibility and Risk Matrix as an Aid

The flexibility and risk matrix introduced in Chapter 5, Section 5.3, can again provide you with objective assistance in considering whether you should revise your targets or KPIs in the quality area. The more inflexible your product is—either due to the nature of the product itself or due to regulatory constraints—the higher you should set the quality benchmark. With low flexibility, the downstream revision will be much more complex and expensive than for a system, product, or service with a high degree of flexibility.

Similarly, regarding risk: The higher the risk associated with using your system, product, or service, the higher the quality should be, even if it is a very flexible and customizable product. After all, very few people want to make an adjustment only after the damage has already been done.

Let's take another look at the matrix (shown in Figure 13.1 again): The further down to the left your solution is, the more relaxed you can be. Conversely, the further up and the further to the right you slide, the more tense you should become and the higher demands you should make on the quality of your solution.

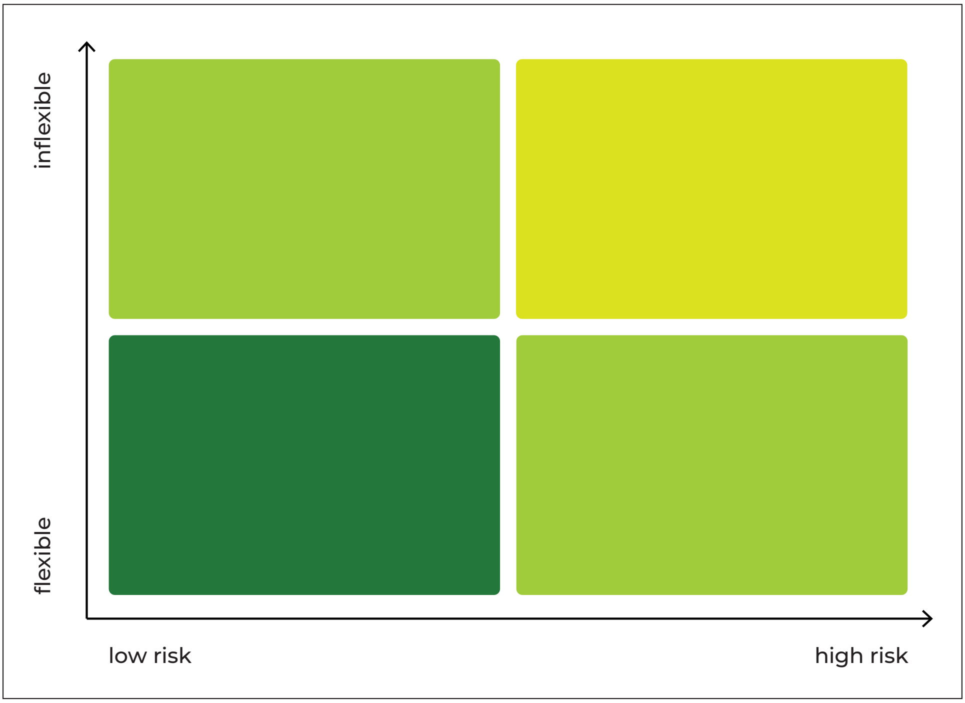


Figure 13.1 The flexibility and risk matrix for determining the required data in the data-driven UX design process.

Therefore, if you find that your target values for quality have been achieved, but you still have areas of high (unmanaged) risk, then you should re-sharpen because that is what the quality targets were designed to prevent. The same is true when you realize your time has run out, but the quality is still insufficient. Sometimes, a quality that can be optimized is better than no product; sometimes, that is not the case. In the next section, we'll share our thoughts on when that is the case.

13.4 Choosing the Right Approach

Now that you have learned about the two most important approaches, we would like to show you in this section which approach we recommend to whom. Our answer comes directly from the criteria. You then decide based on quality if the quality is the more critical good to you, and you take time as a criterion if time is the more important good. Helpful, isn't it?

Okay, okay, that was still very general. To be more specific, choose an end by the *quality* if:

- You have an established solution and want to introduce a successor solution.
- You own an established brand and want to introduce a new solution under that brand.

So, in both cases, you have something to lose that you've already spent a lot of time, effort, and money building up. A poorly designed system, a non-functioning product, or an unreliable service can ruin much more than just this one business opportunity. Accordingly, in these cases, the focus is on quality. You should only publish anyway in very selected cases (for example, due to a time constraint), even if the quality is insufficient.

An example of such an exception is the COVID-19 pandemic. So many new challenges had to be addressed quickly that developing pragmatic solutions was better than developing solution with fewer bugs but only on the market a year later. However, as you can see from the chosen example, we believe there are very few moments when you should bypass the rules as an established player. We recommend a catch-all strategy to limit the potential damage if you plan to do it anyway. This strategy could be, for example, launching under a different brand name or launching in a closed test user group.

In most other cases, we recommend that you take the *time approach*. Since you're never finished in development anyway but always have to respond anew to the changing tasks, goals, and needs of users and to changing environmental conditions, waiting until everything is perfect rarely makes sense. After all, what's perfect can only be perfect during a brief snapshot in time. In the following cases, we would therefore advise you to set the time when you want to be ready:

- You're a startup, and it is your first product, system, or service.
- You launch a new brand that does not concern you or your existing business.
- A defined, externally set time exists by which you must be ready (e.g., because of a change in the law, a trade fair).
- You're doing something completely new that doesn't exist anywhere else.

As you can see, we assume that time-based planning is preferable in most cases, even for iterative processes. This approach also makes life easier for you in overall project management. After all, you're also planning a development period here. And yes, we are also aware that a mixture is often applied. You have a schedule and aim to achieve certain quality goals within the schedule. As in many other usability and UX design areas, there is no black or white, but mostly a "both-and." The two orientations indicate where the focus should be in your consideration. But now, let's look at our three examples to find out when the line was drawn here.

13.5 What's Next for moveHUB

Matthew, Emma, and Anthony pursued and developed their moveHUB idea using data-driven UX design. If you asked the three of them today, they would answer that it helped them get to the point they are now at more quickly. If they were to develop a product again, they would spend even less time on the initial technical prototypes and instead focus even more intensively on the users. Perhaps they would have reached the point even more quickly, ultimately leading to them having to rethink their product again. They were not aware until the first user contact that there was an expectation that such an app would be free of charge.

The three were able to solve this problem by redesigning but through a *pivot* (a change in the business model). Now, the users are no longer those who must pay a fee. Instead, they act as intermediaries, bringing customers to existing mobility solutions: If moveHUB arranges a ride on an e-scooter, the founders get a commission. It remains to be seen whether this will work—but at least the representatives of the mobility providers are open to talks. We're keeping our fingers crossed for Matthew, Emma, and Anthony.

13.6 What's Next for BusinessBooster

Markus Meier's team has reached some critical milestones on the way to BusinessBooster 2.0. Markus is enthusiastic about the "new way of developing" and is sure the successful product will be successful in the end. He also draws this certainty from the fact that regular customers are enthusiastic about the measures and the usability and UX tests that have been carried out. Initially, Markus assumed it was impossible to approach customers with a half-finished product. This opinion has changed completely. He has found a precious "development service provider" in his regular customers. The opportunity to give feedback and be involved in further development has meant that some shaky customers have not bailed out.

In the meantime, *BusinessBooster 2.0* has gone live as a beta. Some customers are involved in the test program, and others are subscribed to a long waiting list. For Markus, fixing the last bugs and then having a big relaunch is still a matter. We hope to be invited to the party.

13.7 What's Next for Healthyfy

For Linda, the big day is just around the corner. For almost 18 months, the team has given everything to turn Healthyfy into a product. The many usability and UX tests have paid off: Linda is almost sure that the final presentation to the decision-makers at the parent company will be a great success. After all, the data collected will enable her to show that patients and physicians specializing in diabetes would use Healthyfy in the final implementation and that a valid business model could be set up. Finally, she wants to use visually designed prototypes to convince stakeholders with an outstanding product design.

Her key message is, "We have the finished blueprint and can show with data that it works. Now we 'just' need to program it and sell it wholesale." We think these are dream conditions for investing.

Chapter 14

Decision Matrix for Method Selection

> *"If you want to have good ideas, you must have many ideas. Most of them will be wrong, and what you have to learn is which ones to throw away."*
> *—Linus Pauling (Nobel Laureate in Chemistry and Peace)*

We introduced you to many different methods that you can use for usability and user experience (UX) design in the last few chapters. So that you don't need to read an entire book each time you need to select the proper method for your problem, you'll find a handy decision matrix for method selection at the end of this chapter (shown in Figure 14.1). This matrix provides an overview of all the methods listed in the book, classified according to their suitability for different questions.

Since generalizations about method selection are challenging to make across the board, we had to make two preliminary assumptions to make this comparison possible:

- **Interviews**
  Whenever we refer to *interviews*, *group interviews*, or *quantitative surveys* in the decision matrix shown in Figure 14.1, we assume that you're interviewing users from the actual user group but that they cannot try out the system, product, or service in the context of the method mentioned. At most, they might be shown pictures but otherwise cannot interact with your design. We realize you can use interviews in other ways, and combining many methods is also possible. We have chosen one variant here for the evaluation.
- **A/B testing**
  We proceeded in exactly the same way with the A/B testing. As we described in Chapter 12, Section 12.10, that A/B testing can be performed as a particular form of usability and UX test or as a quantitative, partially automated test. In the decision matrix shown in Figure 14.1, we refer to the quantitative and partially automated variant and integrate the A/B comparison into the "standard" usability and UX test.

We then compiled various questions that development teams have brought to us. We then rated how well each method answered each question, with three rating levels:

- A green circle: The method can answer the question comprehensively.
- A half-filled green circle: The method can answer the question only in parts.
- An empty circle: The method cannot answer the question.

An application example: If you want to answer something like "How and where is my application used?" read the appropriate bullet in this chapter.

In the matrix, you search out all methods with a filled circle in the associated column: "Field Observation" and "Contextual Inquiry." In addition, with limited suitability, the methods "Secondary Research/Desk Research," "Interview/Jobs-to-be-Done," "Group Discussion," and "Diary Study."

If you want to answer several questions simultaneously, use several columns. If you find a method that fits all questions, use it. If you think you need to use more than one method, select a combination of several methods marked with filled circles.

To make the table and the questions easier to understand for you right away, we have included an overview of the topic areas, with brief explanations and three possible questions each. We hope this outline will help you define your project and your own questions; the questions we provide are only examples, and many more questions could be asked in each topic area, which would then also set your focus if necessary. Therefore, consider the following classification and example questions as a guide that you can use and expand as needed:

- **Identifying user groups**
  Knowing one's users and their characteristics and differentiating them from non-users is particularly important in the design of outstanding products. Typical questions in this area include the following:
  - Who is using my solution?
  - For whom is my offer attractive?
  - What distinguishes the users of my solution?
- **Understanding general requirements**
  Some requirements on a product result from the product's design, but other requirements may result from the task itself and/or the organizational and situational framework conditions. These rather general requirements are the subject of the following questions:
  - What do users generally expect from a solution?
  - What progress must a successful solution bring?
  - What framework conditions must be taken into account when designing the solution?
- **Understanding social interactions**
  When designing solutions, people often think about users and their tasks. However, the quality of the solution is also determined by whether it adequately takes into account existing social interactions and even uses them beneficially, if necessary. In this regard, you need to understand what social interactions exist in the context of using your application. Typical questions in this area include the following:

  - Will my solution be used alone or shared with others?
  - How does the presence of others affect usage?
  - Does usage behavior change when other people are present?
- **Understanding user needs and tasks**
  The best solutions meet the users' needs and support them in performing their tasks successfully. To achieve this goal, you need to know their needs and tasks. Typical questions in this area include the following:
  - What do my users expect from a solution?
  - What do the users want to achieve?
  - What are my users afraid of?
- **Assessing the importance of needs and tasks**
  After you know what needs are relevant, an essential task is to understand which needs and tasks are most important so that you can focus on them in the design. Questions that come up in this area include the following:
  - What needs are most important in the context of my application?
  - What needs should I focus on when developing my application?
  - What would make my users choose my solution?
- **Asking for opinions**
  Especially in the very early stages of product development, understanding how your users feel about a particular topic is often essential. With this knowledge, you can better decide on a focus for the solution and estimate the probability of success of your idea. Frequently asked questions in this area include the following:
  - What are the attitudes of my users?
  - How do my users feel about a particular issue?
  - What arguments do my users use to represent their point of view?
- **Presenting knowledge about users**
  You can gather knowledge about your users, their usage environment, and tasks through many methods. To use this knowledge in development and design, you must prepare it for yourself and your team. We encounter the following questions again and again in this phase:
  - How can I easily talk about my users with others?
  - Who are my users, and what are their core characteristics?
  - How can I show colleagues for whom the solution must be suitable?
- **Understanding the context of use**
  The context of use defines, in large part, what distinguishes a good solution from a bad solution. Questions we encounter in this area include the following:

  - How and where will my application be used?
  - In what environment does the usage take place?
  - What special features must be taken into account during use?
- **Getting to know processes**
  Suppose you want to ensure in the design that your solution optimally adapts to the existing processes or even further optimizes them from the user's perspective. In this case, a fundamentally important task to know the current processes. This understanding includes asking the following questions, among others:
  - How is my application currently being used?
  - How do users solve your problem right now?
  - What steps do my users take to complete the task?
- **Developing an understanding of processes**
  Knowing the users' processes is one step; understanding the processes and knowing why something is done in a particular way is quite another. It is this understanding that enables you to design new and better processes. Typical questions to ask in this area include the following:
  - Why do my users behave one way or another?
  - Why do my users choose to act contrary to the guidance?
  - How is it that certain mistakes are made over and over again?
- **Presenting knowledge about processes**
  Once you know the processes and their specifics, you need to prepare the knowledge about them in a way that makes it easily accessible to you and your team. Questions in this context include the following:
  - How can I easily represent the existing processes?
  - How can I easily discuss procedures with others?
  - How do the processes look in summary?
- **Evaluating existing solutions**
  When you start a new development, you must understand how current solutions perform and what is good and less good about them. This understanding applies to competing and existing products of your own. Questions in this area include the following:
  - How do my users evaluate an existing solution?
  - Where does my application stand in favor of the users?
  - What is the overall impression of my solution so far?
- **Identifying problems in the use of solutions**
  For most problems, your users probably already have a solution. Suppose you want to place your solution successfully. In that case, you need to understand where there are more problems than necessary when using current solutions to optimize them

when designing your own. Questions that come up in this area include the following:

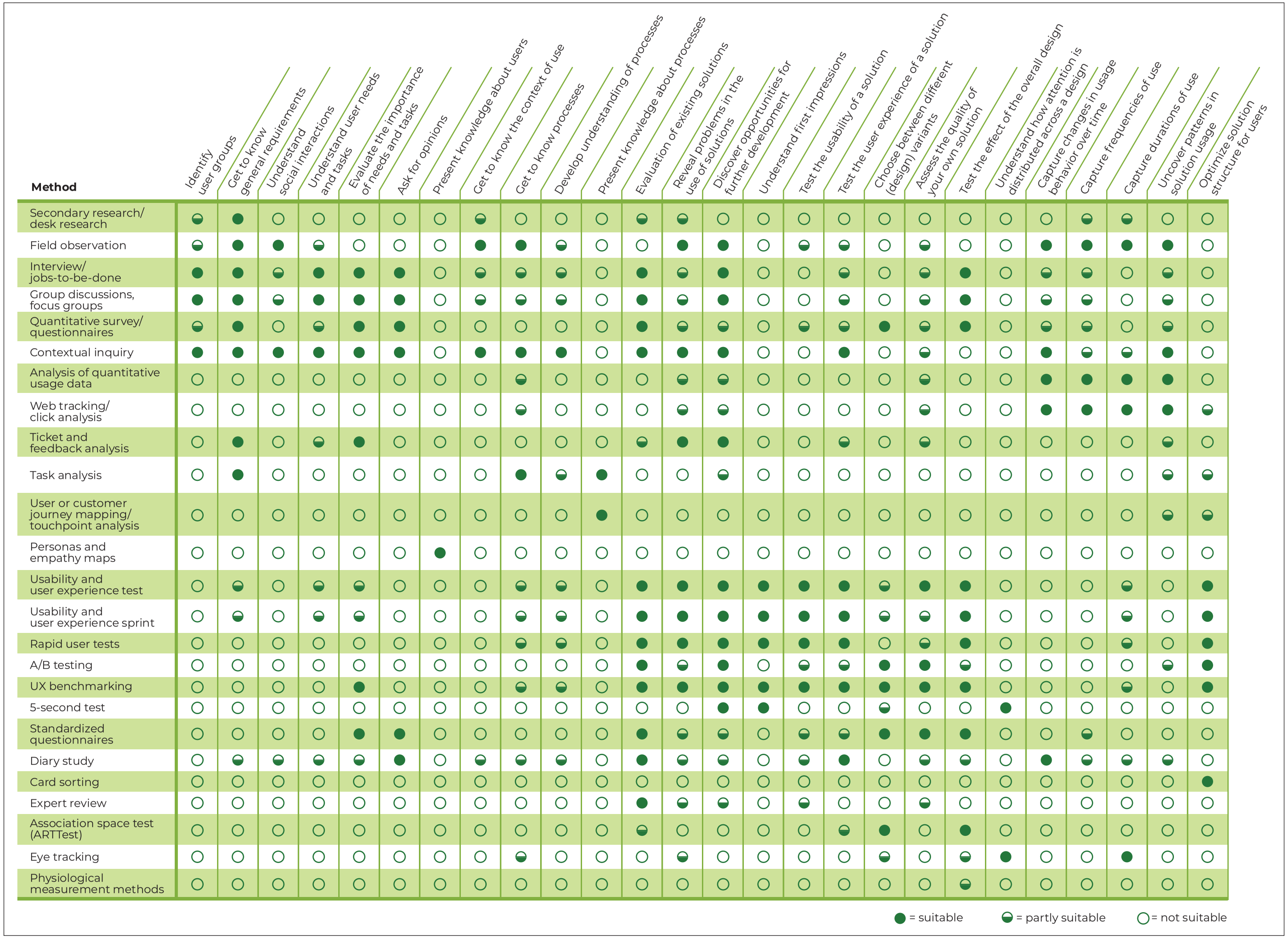
  - Where do my users have difficulties?
  - At what point do my users need more time than necessary?
  - Where can't my users complete tasks successfully?

- **Discovering opportunities for further development**
  The development and release of a solution are significant milestones. However, with the increasing agility of current development projects and the likewise increasing requirements of users, it is crucial that your solution always remains up-to-date and is regularly improved. Questions we encounter in this context include the following:
  - How can I make my product better?
  - Where are there other solutions that do things better than my solution?
  - Where is there potential to develop a better solution?
- **Understanding first impressions**
  There is no second chance for the first impression. This saying is true for people, but it's also true for applications. If you're interested in first impressions, you may be asking yourself some of the following questions:
  - How does my solution look at first glance?
  - What catches the user's eye the most?
  - What is perceived at first glance?
- **Testing the usability of a solution**
  Especially in areas where speed and freedom from errors are essential, testing the usability of a solution is often particularly important. You should deal with this if, for example, the following questions interest you:
  - How easy is my solution to use?
  - Are the users able to use my solution without any problems?
  - Where are the most significant problems when using it?
- **Testing the UX of a solution**
  In addition to pure usability, UX is a key factor in determining the success and acceptance of a subsequent application. Questions that are of importance in this area include the following:
  - What is the UX of my solution?
  - What makes the UX of my solution unique?
  - What has a positive or negative impact on the UX of my solution?
- **Choosing between different (design) variants**
  When designing new solutions, there is no clear right or wrong. That's why there are often different ideas about how a task could be solved, which workflow is the best, or which design looks good. Typical questions in this area include the following:

  - Which variant is the best?
  - Which variant do my users like best?
  - Which variant is best to use?

- **Assessing the quality of your solution**
  In addition to testing individual aspects only on your own solution, understanding how your solution compares to other solutions is often interesting. This comparison may involve predecessor products but also competitive offerings. Questions that aim at an assessment in comparison to others or to benchmarks include the following:
  - How do the users perceive the quality of my solution?
  - How good is our solution objectively?
  - How does our solution compare to others?
- **Testing the impact of the overall design**
  At the beginning of development, the focus is often on the desired effect of the application, and a lot of time and effort goes into defining it. If you want to know whether the desired impact occurs, then you might ask yourself some of the following questions:
  - What is the overall impact of my solution?
  - Does the impact of my solution meet my requirements?
  - Do the users perceive the solution as we have planned it?
- **Understanding how attention is distributed in the design**
  Especially if you're designing an application that is only used to accompany a primary task, or if you want to highlight certain elements specifically, you may be interested in understanding how your application's users' attention is distributed. Some typical questions in this area include the following:
  - Which elements in the interface attract the most attention?
  - Is a particular element in the user interface seen at all?
  - How much attention do users pay to the application as they complete their tasks?
- **Capturing changes in usage behavior over an extended time**
  Many methods are concerned with measuring usage behavior once or evaluating it once. However, most products are used over a more extended period, which is why it can be critical to understand how the evaluation of the solution and its usage changes over time. The following questions may arise in this context:
  - How does the behavior of my users change over time?
  - How does my user's assessment change over time?
  - Are there specific events that trigger changes?

- **Capturing usage frequencies**
  In this area, you would like to know how often your product is used. Typical questions you want answered include the following:
  - How often do my users use my solution?
  - Are there differences in frequency of use between my user groups?
  - Are there changes in the frequency of use at certain times?
- **Capturing usage durations**
  In addition to the frequency of use, another essential aspect of consideration is the question of the duration of use. The following questions are often asked in this regard:
  - How long will my application be used?
  - Are there differences in usage time between my target groups?
  - Can specific triggers be identified that change the duration of use?
- **Capturing usage particularities**
  Sometimes, a necessary and exciting thing to understand is if any particular aspects are highlighted when using your product or comparable products. Typical questions in this area include the following:
  - Are there things that are special about using my application?
  - Are there things that are special in my context of use?
  - Are there situations that lead to a specific behavior in use?
- **Discovering patterns in the use of solutions**
  Especially when optimizing products, attracting new target groups, or identifying new application areas, a beneficial activity is looking for patterns in usage. Questions that arise in this area include the following:
  - Can I discover different groups based on usage patterns?
  - Are there certain situations in which my users exhibit the same behavior repeatedly?
  - Are there typical paths that my users take?
- **Optimizing the solution structure for users**
  Especially in complex applications with many different contents, finding a structure and arrangement of the content that fits users' expectations can be difficult. Questions that arise in this area include the following:
  - How can I increase the findability of content?
  - Where would my users expect to find a particular piece of content?
  - What should the individual menu items be called so users can find them easily?



Method	Identify user groups	Get to know general requirements	Understand social interactions	Understand user needs and tasks	Evaluate the importance of needs and tasks	Ask for opinions	Present knowledge about users	Get to know the context of use	Get to know processes	Develop understanding of processes	Present knowledge about processes	Evaluation of existing solutions	Reveal problems in the use of solutions	Discover opportunities for further development	Understand first impressions	Test the usability of a solution	Test the user experience of a solution	Choose between different (design) variants	Assess the quality of your own solution	Test the effect of the overall design	Understand how attention is distributed across a design	Capture changes in usage behavior over time	Capture frequencies of use	Capture durations of use	Uncover patterns in solution usage	Optimize solution structure for users
Secondary research/ desk research	◐	●	○	○	○	○	○	◐	○	○	○	◐	◐	○	○	○	○	○	○	○	○	○	◐	◐	○	○
Field observation	◐	●	●	◐	○	○	○	●	●	◐	○	○	●	●	○	◐	◐	○	◐	○	○	●	●	●	●	○
Interview/ jobs-to-be-done	●	●	◐	●	●	●	○	◐	◐	◐	○	●	◐	●	○	○	◐	○	◐	●	○	◐	◐	○	◐	○
Group discussions, focus groups	●	●	◐	●	●	●	○	◐	◐	◐	○	●	◐	●	○	○	◐	○	◐	●	○	◐	◐	○	◐	○
Quantitative survey/ questionnaires	◐	●	○	◐	●	●	○	○	○	○	○	●	◐	◐	○	◐	◐	●	◐	●	○	◐	◐	○	◐	○
Contextual inquiry	●	●	●	●	●	●	○	●	●	●	○	●	●	●	○	○	●	○	◐	○	○	●	◐	◐	●	○
Analysis of quantitative usage data	○	○	○	○	○	○	○	○	◐	○	○	○	◐	◐	○	○	○	○	◐	○	○	●	●	●	●	○
Web tracking/ click analysis	○	○	○	○	○	○	○	○	◐	○	○	○	◐	◐	○	○	○	○	◐	○	○	●	●	●	●	◐
Ticket and feedback analysis	○	●	○	◐	●	○	○	○	○	○	○	◐	●	●	○	○	◐	○	◐	○	○	○	○	○	◐	○
Task analysis	○	●	○	○	○	○	○	○	●	◐	●	○	○	◐	○	○	○	○	○	○	○	○	○	○	◐	◐
User or customer journey mapping/ touchpoint analysis	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	◐	◐
Personas and empathy maps	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Usability and user experience test	○	◐	○	◐	◐	○	○	○	◐	◐	○	●	●	●	●	●	●	◐	●	●	○	○	○	◐	○	●
Usability and user experience sprint	○	◐	○	◐	◐	○	○	○	◐	◐	○	●	●	●	●	●	●	◐	◐	●	○	○	○	◐	○	●
Rapid user tests	○	○	○	○	○	○	○	○	◐	◐	○	●	●	●	●	●	●	○	◐	●	○	○	○	◐	○	●
A/B testing	○	○	○	○	○	○	○	○	○	○	○	●	◐	●	○	◐	◐	●	●	◐	○	○	○	○	◐	●
UX benchmarking	○	○	○	○	●	○	○	○	◐	◐	○	●	●	●	●	●	●	●	●	●	○	○	○	◐	○	●
5-second test	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○	○	◐	○	○	●	○	○	○	○	○
Standardized questionnaires	○	○	○	○	●	●	○	○	○	○	○	●	◐	◐	○	◐	◐	●	●	●	○	○	◐	○	○	○
Diary study	○	◐	◐	◐	◐	●	○	◐	◐	◐	○	●	◐	◐	○	◐	●	○	◐	◐	○	●	◐	◐	◐	○
Card sorting	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
Expert review	○	○	○	○	○	○	○	○	○	○	○	●	◐	◐	○	◐	○	○	◐	○	○	○	○	○	○	○
Association space test (ARTTest)	○	○	○	○	○	○	○	○	○	○	○	◐	○	○	○	○	◐	●	○	●	○	○	○	○	○	○
Eye tracking	○	○	○	○	○	○	○	○	◐	○	○	○	◐	○	○	○	○	◐	○	◐	●	○	○	●	○	○
Physiological measurement methods	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	◐	○	○	○	○	○	○

● = suitable ◐ = partly suitable ○ = not suitable

Figure 14.1 Decision matrix for method selection. A full PDF of this image is available for download at www.rheinwerk-computing.com/5778 under the product supplements.

Click image to view horizontally

Chapter 15

Wrap-Up and Conclusion

"When the others think you're at the end, you have to really start."
—Konrad Adenauer (first chancellor of West Germany)

This book contains an overview of the core topics of usability and user experience (UX) design, presented as a mix of theory and practice. We hope the book can support you in getting your picture of the discipline. Since the volume is now quite considerable, at more than 600 pages, we would like to summarize for you once again at the end the six core messages of the book, which form the essence of the whole.

15.1 It's All about the Users

Probably the most crucial aspect of this book is focusing on the users of your system, product, or service. When a human uses and operates your solution, you need usability and UX design. The technology you implement plays an important role, but a product is worthless if it doesn't add value to its users. That's why our first wrap-up message is:

> *Message 1: Usability and UX design is always about the users.*

In this book, we provided an overview of a wide range of methodologies that you can use to develop outstanding solutions. In Chapter 14, we again showed you several methodologies and guided you on how to choose among them. However, as you probably know by now, you can adapt or combine many methods, so the choice of the ultimate method, while important, is not "decisive in the war." Only one exception to the rule exists: If you're developing a new product and have not collected feedback from your users at least once during the entire development process, you have done something fundamentally wrong. Of course, gathering feedback more often and, more importantly, regularly is better. Usability and UX design is all about the following:

1. Understanding who your users are and what they need
2. Translating this knowledge into a design that meets the requirements
3. Testing whether this design worked for your users
4. Based on the result of testing, adjusting the design until it fits

If you stop reading now, you have taken the most important thing with you. Design for your users; accordingly, only the users can tell you whether your design succeeded.

15.2 Usability and UX Design Is Interdisciplinary

It's great if you're a lone wolf on your company's usability and UX design journey. Even that first hire is a big step, especially if the company has never before dedicated itself to the topic. Overall, however, usability and UX design require different expertise. Our wrap-up message is, therefore:

> *Message 2: Usability and UX design is an interdisciplinary field for which you need many different skills.*

These skills include the following:

- Knowledge of the suitable methodology
- The ability to conduct sound research
- An understanding of the technology and a good interaction concept
- A good design sense and creative skills for the final design

Only a few UXers we know are brilliant in all areas. Complete competency in all areas is not necessary, either. You just need to be aware that all aspects are part of it and that you may need support.

15.3 UX Is Not Designed, It Is Experienced

Indirectly, this message is already found in the first message in Section 15.1. Still, we want to re-emphasize: Even though this book is called *Usability and User Experience Design*, we're designing the UX of your users. You design a system, a product, or a service, and this solution subsequently triggers a UX for your users. The UX is always subjective and depends on the perceptions of users. The wrap-up message we want to give you in this context is thus:

> *Message 3: You design the system, the product, or the service. UX cannot be designed, only measured.*

Thus, to determine whether you have designed a solution with a good UX, there is no way around talking to your users and testing that—one way or another.

15.4 The Path to an Outstanding Product Is Iterative

With the understanding that users are the measuring stick for the success of your design (and, of course, our design) comes the knowledge that a theoretical assessment of product quality and a theoretical collection of product requirements can never be complete. Likewise, five users cannot tell you all the important points you need to know for the overall development of your product. You implicitly agree that further iterations are expected by first designing a prototype and testing it in a usability and

UX design process. You design a solution. You test it with users. You learn from it what needs to be better. And then you start all over again. This results in the following wrap-up message:

> *Message 4: The path to a successful system, product, or service is iterative and always follows the idea of testing our design and adapting it to test results.*

The cycle can be planned but not abbreviated. You can't throw ten times as many participants into the first test and assume you're done. The feedback you get leads to a new design. The new design leads to new feedback. Whether you like it or not, usability and UX design is iterative. You can time the end of the process for the first release or time it based on the quality you want. But no matter what you do, more than one iteration will occur.

15.5 Once You Start with Usability and UX Design, You Never Stop

In the development of every system, every product, and every service, there comes a time when usability and UX design must begin. And the sooner this moment comes, the better! In agile development and with digital products, however, you should be aware that usability and UX design never ends.

> *Message 5: Choosing a human-centered development process like data-driven UX design means viewing your system, product, or service as "permanently under development" and never stopping with usability and UX design.*

Once you design in a human-centered way, this process is never-ending and stays with you even after your solution is released. You'll receive feedback and support tickets. You'll receive feedback on social media. You can plan updates and integrate new features, maybe even launch a follow-up version in the foreseeable future. All of these opportunities are accompanied by usability and UX design so that your solution is not a one-off that is good only in the short term but instead remains outstanding in the long term. Usability and UX design is not a project but a way of thinking and developing. It is the commitment that makes your solutions successful in the long run.

15.6 Design Based on Data

Our final message is concise but very important. Use data to design your systems, products, and services. Don't rely on your gut feelings and experience: Generate data from, with, and about your users and the usage of your solution and then use that data. Data-driven UX design is the path to success. Our final wrap-up message for you is accordingly:

> *Message 6: Let data guide you through the design process to ensure you're on the right track.*

15.7 Closing Words and Call for Participation

We’ve come to the end—of course, only in terms of this book’s first edition. We hope you had as much fun reading this book as we had fun writing it.

When you write such a comprehensive book, one that is supposed to give an overview of an entire subject area, on one hand, a lot of work goes into creating the content. On the other hand, the setting of the content also shows one’s focus. Currently, this book is our book, but we want to make it your book over potential subsequent editions. Let us know which topics and focal points you want us to consider for the next edition. What did we miss? What was too much? What do you see differently? We welcome your feedback. Feel free to e-mail us at *handbook3DUX@custom-interactions.com*.

In addition, we would love to know about the experiences and results you achieve with these approaches and our suggestions. Feel free to keep in touch with us as well on LinkedIn at *https://www.linkedin.com/in/dr-michaela-kauer-franz/* and *https://www.linkedin.com/in/dr-benjamin-franz/*.

Thanks for reading!

—Ben and Michaela

Appendices

A Glossary 639
B Bibliography 653
C The Authors 663

Appendix A
Glossary

3DUX → Data-driven UX design: Our data-driven user experience (UX) design approach advocates integrating user data and user feedback in as many places as possible and aims to ensure a successful product.

5-second test: A UX research method in which users look at a product for 5 seconds and then share their impressions of the design. This method aims to evaluate which first impressions about the design are formed and what users can understand within this short time.

A/B testing: In this UX research method, users are confronted with different product variants consciously (in the context of a usability and UX test) or unconsciously (in the context of live A/B tests). With A/B testing, you can decide on the better variant.

Act: Act is the fourth information processing step in the stimulus-response model after decision-making. Describes the ability to perform a previously decided action.

Active listening → Listening: Listening style characterized by active acknowledgment of what is being said, summarizing, asking questions, and being approachable (e.g., through eye contact and smiling).

Analysis: Examination in which something is dissected (i.e., the whole is broken down into its constituent parts).

Ask questions: Method of user research to generate essential data for one's product design by asking specific questions:

- *Open questions*: In this case, the respondent is not given any answer options. Open questions are almost always WH-questions (e.g., involving how, when, where, etc.).
- *Closed questions*: In this case, the answer options are limited and predetermined. Two examples of closed questions are scale questions and yes/no questions.

Association space test (ARTTest): The ARTTest is a test procedure that uses a questionnaire to compare the associations desired by the manufacturer with the actual associations of the users for an application, thus enabling control of the design.

Assumption: An assumption is a conjecture or opinion on a subject not supported by facts.

Assumptions-knowledge matrix: The assumptions-knowledge matrix separates knowledge from assumptions and everything in between.

AttrakDiff: A standardized procedure that assesses the pragmatic and hedonic quality of an interactive product from the user's perspective.

B2B: Standing for “business-to-business,” a B2B business model focuses on the sale of goods or services from one company (business) to another.

B2C: Standing for “business-to-consumer,” a B2C business model focuses on the sale of goods or services from a company (business) to end customers (consumer).

Brainstorming: Brainstorming is a creative method for generating ideas. One person or a group of people document and develop ideas, which are “thrown” into the room. The entire group can further develop these ideas.

Brainwriting: Brainwriting is a creative method for finding ideas. Participants individually put their ideas (written and graphic) on paper and then explain them to the group for discussion and further development.

BusinessBooster: A fake company we invented that serves as an example throughout the book. It symbolizes a former market leader that wants to work back up to the top of the market. Its offered product is a customer relationship management (CRM) business intelligence (BI) tool for small and medium-sized enterprises (SMEs) in the B2B sector.

Button: An interactive area on a user interface (UI) hat can be activated by clicking with a mouse or tapping on a touch display to trigger an action.

Buyer → Customer: A person who (potentially) spends money on a system, product, or service.

Call-to-action (CTA): A UI element that calls the user to perform a specific action (e.g., a button, link, or text).

Card sorting: A methodology with which menu structures can be developed and tested with users.

Checkbox: A checkbox is an interactive element of a graphical interface. It indicates whether a particular option has been selected (checked or ticked) or not selected (unchecked or unticked) by the user. Checkboxes are mainly used when one or more options can be selected simultaneously from several alternatives.

Click analysis → Web tracking: The tracking and evaluation of a user’s click patterns when using a (web-based) software product to generate important usability data.

Click dummy: A click dummy is a prototype for user tests. It represents workflows and, if necessary, animations.

Combo box: Combo boxes represent a combination of two different input methods. On the one hand, it is a text field where free text can be written. On the other hand, a combo box is a kind of dropdown where a suitable result can be selected from the entries based on one click.

Conformity with user expectations (interaction principle): The interactive system’s behavior is predictable based on the context of use and commonly accepted conventions in this context. Key questions: Does the system behave appropriately, and does it also provide appropriate feedback to the users? Is the system consistent internally but also externally? Are there any changes in the context of use that would need to be considered?

Context of use analysis: At its core, the context of use analysis is about understanding the current situation. Thus, data is collected on a combination of users, goals, tasks, resources, and environment. This information is then broken down into their components.

Context of use: A combination of users, goals and tasks, resources, and environment (ISO 9241-11:2018).

Contextual inquiry: Structural observation and interview of potential users or potential patients in their environment to understand the environment, the characteristics of the person, possible sources of danger, and individual steps of use.

Controllability → Interaction principle: The interactive system allows the user to maintain control of the UI and the interactions, including the speed and sequence and individualization of the user-system interaction. Key questions: How well does the solution support interruptions by users? How much flexibility does the solution allow users? How much individualization and customization does the solution allow users?

Conversion rate: The percentage of users who perform a desired action divided by the total number of users.

Conversion: The desired action (from the view of the developer/operator) a user of a system, product, or service should do.

Critical function → Main workflow: The main workflow describes the essential workflow or function of a product. In UX design, the main workflow is prioritized.

Customer → Buyer: A person who (potentially) spends money on a system, product, or service.

Customer experience: A person's comprehensive buying experience and associated impressions.

Customer lifetime value: The average revenue a company can expect to generate with a customer over the entire business relationship.

Data-driven UX design → 3DUX: Our user-centric design approach advocates integrating user data and user feedback in as many places as possible to ensure a successful product.

Decide: Decide is the third information-processing step in the stimulus-response model, after recognition. Describes the ability, based on a recognized stimulus, to select an appropriate action from the user's point of view and, thus, a reaction based on the experience.

Design expertise: Skills needed to develop solutions that successfully implement user requirements.

Design grid: Our design grid is a grid that divides the screen into different areas and thus serves as an auxiliary construction for the arrangement of individual design elements.

Design heuristics: Heuristics are procedures and strategies for the targeted solution of problems. They represent empirical values and help to make the best possible decisions with limited knowledge quickly.

Design research: In design research, the visual design of a product or application is first researched in comparison to the customer, the industry, and current trends and then evaluated and tested with users during development.

Design thinking: Design thinking is an approach that aims to solve user problems at the design level in addition to shaping and designing.

Desk research → Secondary research: Obtaining information using existing data sources.

Diary study: When using a finished system, product, or service, people are asked to provide ratings at regular intervals to draw conclusions about the UX.

Direct observation: Observation by a person present on-site.

Discover: The first information-processing step in the stimulus-response model, after the initial stimulus. Describes the ability to filter out a specific stimulus from the abundance of available stimuli.

Double Diamond model: The Double Diamond is a model for describing an ideal approach to human-centered design projects. It is a comprehensive framework that not only describes the design work process but also provides core principles and design methods to guarantee good design work in the long run. The core idea of the Double Diamond is the integration of divergent and convergent thinking.

Double opt-in: This procedure is a double consent procedure for providing one's contact information. For example, users are only added to a newsletter distribution list if they have both entered and confirmed their e-mail address.

Dropdown: A dropdown is an element of graphical interface design. This menu item can be expanded or collapsed by clicking it, thus revealing certain information only by expanding it and making it available for selection. Dropdown menus can be used to make an interface more straightforward.

Effectiveness, efficiency, and satisfaction: These three criteria must be met by a product during its use, according to ISO 9241-11:2018, to exhibit high usability. A product must allow a goal to be achieved effectively, efficiently, and in a satisfying way:

- *Effectiveness*: Describes the accuracy and completeness with which a goal can be achieved.
- *Efficiency*: Describes the extent to which resources are used in relation to the results achieved.
- *Satisfaction*: Describes the extent to which a user's physical, cognitive, and emotional responses that result from the use of a system, product, or service meet their needs and expectations.

Empathy maps → Personas: Standardized representation of characteristics of typical user or customer groups similar to a persona.

Ergonomics: This scientific discipline is concerned with understanding the interactions between people and other system elements. It also refers to the best possible adaptation of working conditions for people.

Expectation management: This term describes all activities of a company that aims to adapt its products, systems, and services to the expectations of its customers to meet these expectations.

Expert evaluation: A group of methods for evaluating a system, product, or service by experts who put themselves in the shoes of the user.

Expert review: UI testing by experts. UX guidelines are used to ensure a consistent evaluation scheme.

Eye tracking: A usability and UX method that tracks eye movements while using a system, product, or service. This method allows not only the user's actions but also the user's attention to be tracked.

Feature requests: The request for missing functionalities of a system, product, or service.

Field observation → Shadowing → On-site visit: Method to collect data for a context of use analysis. Data about people, processes, and the environment is generated by passively observing the user in the actual usage environment.

Flexibility and risk matrix: A matrix on which to locate one's solution in terms of flexibility and risk to address the following questions:

- How flexible is the solution after it has been fully developed and launched?
- What risk does my solution pose in terms of incorrect operation and misinterpretation?

Focus groups → Group discussion: A focus group is a UX method aiming at obtaining the broadest possible spectrum of opinions in the shortest possible time. A discussion guided by a moderator on a given topic. The number of participants is usually between 6 and 12.

Formative evaluation: Testing of a product during development that evaluates usability and/or UX and generates requirements for improving the product.

Frequency of use: The frequency of use is important usage data about a product. Usage frequency provides information about how often a user uses a particular feature.

Friendly user: Users who are in some relationship with the product's developers and therefore do not give open and honest feedback on the product due to bias. For example, when interviewing a friend who is afraid to give negative feedback for fear of offending another person.

Functional scope: Includes all functions of a product.

Gestalt principles: Gestalt principles are a collection of simple laws that support understanding the perception of design. Through them, good decision-making in design is promoted.

Group discussion → Focus groups: A focus group is a UX method aiming at obtaining the broadest possible spectrum of opinions in the shortest possible time. A discussion guided by a moderator on a given topic. The number of participants is usually between 6 and 12.

Guided interview → Qualitative surveys → User survey: A (non-written) survey is a qualitative empirical method to elicit users' perceptions and opinions or user needs and requirements. The survey can be conducted with the help of a guideline but can also be semi-structured, consisting of a guideline and situational questions for more in-depth information. Both individual and group interviews are possible.

Halo effect: The halo effect is a perceptual bias in which individuals infer other traits (e.g., friendliness) based on one trait (e.g., attractiveness).

Hazard-related use scenarios: Dangerous scenarios that arise from a usage error.

Healthyfy: A fake company we invented that serves as an example throughout the book. It symbolizes a spin-off company of a major medical device manufacturer. Healthyfy is a referral portal for healthcare professionals (i.e., a telemedicine application). Healthyfy aims to create a digital platform where doctors from different specialties and patients can log in via a web interface.

HIPPO effect: Standing for the "highest paid person's opinion," HIPPO represents the opinion of the most important person in the room. Refers to the effect that the opinion of the highest-paid team member or the person with the highest position in the hierarchy is often given the most attention, even if far better solution proposals exist on closer examination.

How-might-we questions: The term comes from design thinking and pursues the goal of formulating the mission in a single question. This question makes it much easier to focus on and question the core topic during the project.

Human factors: The term encompasses various psychological, cognitive, and social factors influencing humans on socio-technical systems.

Human-centered design: In this design process, the future users of the solution with their needs and requirements are placed in the focus of the development, and a suitable design is created with the help of knowledge in the areas of ergonomics, human factors, and usability.

Human-machine interaction, human-computer interaction: By human-machine interaction, we mean any exchange of information between humans and machines. This information can trigger actions (e.g., operating procedures, human reaction) but does not have to. We also consider human-machine interaction to be information that conveys status information between the human and the machine without a direct request for action.

Icon: An icon is a graphic symbol (e.g., in applications, files, etc.). In contrast to pictures and graphics, icons are usually highly abstract and simplified.

Information processing: Describes a person's process when processing a stimulus toward a response.

Interaction principles: A collection of general principles for the design of human-machine interfaces from ISO 9241-110.

Interview → User interview: A UX research method. A guided conversation to evaluate user needs. Interviews can be structured, semi-structured, or open-ended.

ISO 9241-110 → Interaction principles: An international standard that sets clear guidelines for human-computer interaction design.

Iteration: An iterative process is a repetitive process that aims to improve something further and thus approach an optimal solution. In the UX field, this refers to the iterative testing of product prototypes, which aims to generate improvement potential for the system, product, or service in each iteration.

Kano model: The model helps to prioritize product features in the dimensions of "customer satisfaction" and "expectation." It differentiates between threshold, performance, excitement, and rejection features. It also describes the degeneration of excitement to threshold or rejection features over time.

Key performance indicator (KPI): KPIs are indicators used to measure the degree to which specific targets have been met. This measure can determine the effectiveness of certain efforts and achievements.

Kick-off workshop: A kick-off is a project team meeting at the beginning of a project. In this meeting, basic questions are clarified, and the further project plan is presented.

Knowledge: The totality of attainments that someone has (in a particular field).

Law of closure (Gestalt principle): The observer completes the missing components of a "closed" shape.

Law of common fate (Gestalt principle): Design elements that change or move together are perceived as belonging together.

Law of continuity (Gestalt principle): If design elements appear as a continuation of previous elements, they are perceived as belonging.

Law of figure/ground (Gestalt principle): Design elements in the foreground have a clear shape and boundary. Elements in the background are rather formless.

Law of proximity (Gestalt principle): Design elements that are close to each other are perceived as belonging together.

Law of similarity (Gestalt principle): Visually similar design elements are perceived as belonging together.

Law of symmetry (Gestalt principle): If the attributes of a design element are in equilibrium, then it is perceived as symmetrical (balanced).

Law of the point of view (Gestalt principle): If a design element significantly differs from the others, it attracts attention.

Law of visual hierarchy (Gestalt principle): The size and shape of objects influence the visual importance of the design elements.

Learnability (interaction principle): The interactive system supports discovery of its capabilities and how to use them, allows exploration of the interactive system, minimizes the need for learning, and provides support when learning is needed. Key questions: How well can users discover the information and controls they are looking for? How easily can users safely explore the discovered information and controls? How much does the solution support users in learning the consequences of their interactions?

Likert scale: A Likert scale is a specific scale type for questionnaires that makes it possible to standardize questionnaires completely. Participants answer the questions using values expressing their agreement or disagreement with a statement.

List of requirements → Requirements specifications: Defines an enumeration of what a product must be like and what it must do for the users to achieve their goals in using the product.

Listening → Active listening: Listening style characterized by active acknowledgment of what is being said, summarizing, asking questions, and being approachable (e.g., eye contact and smiling).

Main workflow → Critical function: The main workflow describes the essential workflow or function of a product. In UX design, this is prioritized.

Medical device: A product that, according to its intended use, is used for medical purposes (i.e., for the detection, treatment, or prevention of diseases).

Memory: The ability to remember what has already been perceived, to recall what has been previously learned, and to store and retrieve information.

Mindset: The attitudes and expectations with which a person approaches a subject.

Mood board: A collage on which can be picked up various images, colors, shapes, and texts that can be considered seminal to a visual style.

moveHUB: A fake company we invented that serves as an example throughout the book. It is symbolic of small start-ups that develop a new application from scratch, in this case, a mobility app.

Multiple choice questions: In a multiple-choice question, a question is asked, and several answer options are given. These questions are very suitable for standardizing questionnaires.

Minimal viable product (MVP): The MVP process aims to develop a product that consists exclusively of the core features that are needed to be valuable to customers. Everything else is left out.

Net Promoter Score (NPS): A key figure that expresses whether a customer would recommend a service or product to others.

Objectives (according to ISO 9241-112018): Part of the context of use analysis: what things do users want to achieve by using the product?

Objectivity: A quality criterion for tests. Describes the fact that a statement is free of subjective assessments and evaluations. A statement is objective if several independent persons come to the same conclusion.

Observations: A method in which the context of use and usage patterns are identified and systematically documented through user observations in a realistic or real work environment.

- *Open observations*: In an open observation, the participants know they are under observation.
- *Covert observations*: In a covert observation, the participants do not know they are under observation.

On-site visit → Shadowing → Field observation: Method to collect data for a context of use analysis. Data about people, processes, and the environment is generated by passively observing the user in the actual usage environment.

Pain point and gain point: An area of an empathy map that describes the points that give a persona frustration (*pain*) or pleasure (*gain*).

Patient: A person who seeks medical treatment.

Perception: Perception is the ability of a person to perceive stimuli and process them in a meaningful way. It includes the detection of stimuli and their recognition.

Personas → Empathy maps: Personas are prototypical representations of typical user or customer groups. A more basic version but highly structured are empathy maps.

Primacy-recency effect → Serial position effect: Memory effect that causes information to be better retained and freely reproduced when placed at the beginning or end. Also called the "serial position effect."

Problem-solution fit: The adaptation of certain product features to the problem of a user group by the developer.

Productivity: An indicator of the relationship between output and workload.

Proof of concept: The proof, provided by a test, that a concept also works according to the target.

Prototype: In the field of data-driven UX design, a prototype is a fast and simulated implementation of a solution idea to map the interaction of user and product as quickly and efficiently as possible for testing.

- *High-fidelity prototype*: A prototype that depicts the mode of action, interaction, and designs close to the finished product.
- *Low-fidelity prototype*: A prototype that can be produced quickly and cheaply to make an interaction or mode of action experienceable.

Qualitative surveys → User survey → Guided interview: A (non-written) survey is a qualitative empirical method to elicit users' perceptions and opinions or user needs and requirements. The survey can be conducted with the help of a guideline but can also be semi-structured, consisting of a guideline and situational questions for more in-depth information. Both individual and group interviews are possible.

Quality management system: According to ISO 9001, a quality management system is a method of corporate management to ensure systematic quality management.

Quantitative survey: A quantitative survey is a quantitative empirical method of inquiring about users' perceptions and opinions or user needs and requirements. Quantitative surveys are mainly conducted using questionnaires (online or offline) or telephone interviews using structured questionnaires.

Questionnaire design: Includes all elements of a questionnaire that need to be designed: from the type of questionnaire to the list of questions and the structure.

Rapid user tests: Simplified test procedure in which users can evaluate URL-based content in a largely automated and unmoderated manner.

Rapport: Direct contact between questioner and interviewee, leading to successful cooperation.

Reactance: Reactance is the psychological term for inner resistance to restrictions on freedom of action caused by prohibitions or external pressure. Reactance promotes the tendency to do what is forbidden or undesirable.

Reaction: Result of information processing in the stimulus-response model after the "action" step. Describes the executed action's observable result and represents the user's feedback loop.

Recognize: Recognizing is the second information processing step in the stimulus-response model after discovering. Describes the ability to understand and attach an appropriate meaning to a discovered stimulus.

Redesign: A remake of an existing design.

Remote testing: Testing that is not performed on-site. Remote testing usually takes place via online meeting tools.

Requirement: A requirement specifies what a product must be like and what it must do for the users to achieve their goals when using the product.

- *Range requirement—maximum requirements*: A maximum requirement specifies how high specific values can be. An example would be a price maximum.
- *Range requirement—minimum requirements*: Minimum requirements define what a product must achieve as a minimum.
- *Fixed requirement*: Defines an exact value that a product must achieve.

Requirements specification → List of requirements: Defines an enumeration of what a product must be like and what it must do for the users to achieve their goals in using the product.

Responsive design: Design solutions that allow the UI to be automatically resized and adapted to the resolution of the end device. For example, an interface is adapted to a computer screen, a tablet, and a cell phone display.

Return on investment (ROI): The return on an entrepreneurial activity measured in terms of success concerning the capital invested.

Risk assessment (according to ISO 14971): Risk assessment is the interaction of risk analysis and risk evaluation. In this process, all available information is used to identify hazards and assess them in terms of their probability of occurrence of the damage as well as the severity of the damage. Depending on the result, the manufacturer may be forced to implement risk-minimizing measures for its product.

Risk classification: The consideration of potential hazards and hazard-related use scenarios.

Risk management: ISO 14971:2020 specifies that risks of medical devices must be identified and controlled. The aim is to eliminate unacceptable risks.

Risk: The amount of damage multiplied by the probability of occurrence of the damage.

Root cause analysis: The process of identifying causes of failure during use so that they can be eliminated in the future through product design.

Scale rating: Giving opinions or feedback by assigning numbers. For example, rating one's satisfaction on a scale of 1 to 10, where 1 represents "very dissatisfied" and 10 represents "very satisfied."

Screener: A document full of questions that allow the person asking the question to decide whether or not the respondent belongs to a user group.

Screen flow: A low-fidelity prototype that maps a workflow by stringing together wireframes.

Secondary research → Desk research: Obtaining information using existing data sources.

Self-descriptiveness → Interaction principle: The interactive system presents appropriate information, where needed by the user, to make its capabilities and use immediately obvious to the user without unnecessary user-system interactions. Key questions: Is all the necessary information present and obviously available? Is it clear to users what stage the process is at or what state the solution is in?

Serial position effect → Primacy-recency effect: Memory effect that causes information to be better retained and freely reproduced when placed at the beginning or end. Also called the "serial position effect."

Shadowing → Field observation → On-site visit: Method to collect data for a context of use analysis. Data about people, processes, and the environment is generated by passively observing the user in the actual usage environment.

Simulation: In the UX space, replicating real-world scenarios for user testing.

Slider → Web page slider: A slider is an element that displays information that changes in a certain time rhythm and can also be actively controlled by the user. Sliders are often used as a starting element on web pages to attract attention.

Software ergonomics: Software ergonomics aims to design software products according to the people working with them. Their contents are questions of the dialogue organization, work contents, and the preservation and organization of decision areas.

Stakeholder workshop: Workshop before the start of a project and during the project to inquire about and define stakeholder attitudes, expectations, and goals and to agree on how to proceed.

Stakeholders: Persons who have a direct or indirect interest in the outcome of the development and have a legitimate interest in influencing the development.

Stimulus: A stimulus is an external or internal influence on the organism, e.g., on the sense organs, which triggers a specific reaction not controlled by the will.

Stimulus-response model: The stimulus-response model is a model that shows the steps from stimulus to response: discovery, recognition, decision, and action. The simplified stimulus-response model of human information processing is based on the idea that information processing proceeds in individual sequential stages. Only when one stage has been completed can the next stage start.

Style guide: A document that contains information about what elements exist, what they look like, and how they are arranged.

Suitability for the user's task → Interaction principle: An interactive system is suitable for the user's tasks when it supports the users in the completion of their tasks. Key questions: Is the solution even suitable for solving a specific task? How much effort is needed to accomplish this task(s)? Are there defaults that help users successfully perform the task?

Summative evaluation: A summative evaluation is the final evaluative test of a system, product, or service. In medical technology, summative evaluations are exclusively related to the safety of medical product use. In marketing, it is also used to obtain statements for advertising.

Support ticket: A support ticket allows users of a software or service to report a problem to a service representative.

Survey guide → Test guide: A survey guide is a list of questions for a user survey that serves as a guide to ensure that all important questions are asked.

System Usability Scale (SUS): The SUS is one of the oldest standardized methods for assessing subjective usability. The questionnaire consists of ten statements that are answered on a 5-point scale. Each statement can be answered with a value from 0 = "do not agree at all" to 4 = "agree completely."

Task analysis: Method for decomposing the user task into individual sub-steps with subsequent analysis.

Task expertise: The user's knowledge of the task they are trying to solve.

Ticket/feedback analysis: Investigating user feedback or support tickets for causes of errors or user problems when using a product.

Test guide → Survey guide: A survey guide is a list of questions for a user survey that serves as a guide to ensure that all important questions are asked.

Touchpoint analysis → User (or customer) journey mapping: A user or customer journey map displays the individual stages of a user's or customer's activity. In this way, the user's or customer's journey can be divided into individual stations. Here, one can interpret connections or find individual contact points with one's company. The presentation can have different foci so that the user's journey or the various contact points of the user with the offer are highlighted.

Trial-and-error process: A process of generating errors through experimentation with different methods to learn from them.

Typography: The design of fonts. Essential for the readability of texts. It concerns font size and font type, among other things.

Usability: The standard ISO 9241-11 defines usability as "*The extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.*"

Usability and UX design: An activity that deals with designing systems, products, and services to achieve high usability and an outstanding UX.

Usability and UX designer → UXer: Usability and UX designers translate user requirements into design solutions.

Usability and UX testing: A test to evaluate a UI with the correct user group in a specified usage environment in terms of usability and UX.

Usage data: Data collected during the actual use of the product and available independent of any additional interaction with the users.

Usage environment: The environment in which the use of a product takes place. Included are the localities, the physical nature of the environment, bystanders, and social attributes of the environment, such as elevated stress or noise level.

Usage requirements: Define what a system, a product, or a service must later achieve so that the users can use it or them purposefully.

Use case: One of the most common forms of documenting requirements is the creation of use cases. Use cases are scenarios that can occur when users operate a system, product, or service. Typically, they describe the behavior of an existing or fictitious solution that is visible to the users.

Use error robustness (interaction principle): The interactive system assists the user in avoiding errors and, in case of identifiable errors, treats them tolerantly and assists the user when recovering from errors. Key questions: Does your solution help users avoid errors? Is your solution tolerant of errors made? Does your solution help users fix usage errors and/or mitigate their consequences?

Use error: A use error is an action or lack of action by a user in using a product that results in the outcome of the use being other than that intended or expected by the user.

User (or customer) journey mapping → Touchpoint analysis: A user or customer journey map displays the individual stages of a user's or customer's activity. In this way, the user's or customer's journey can be divided into individual stations. Here, one can interpret connections or find individual contact points with one's company. The presentation can have different foci so that the user's journey or the various contact points of the user with the offer are highlighted.

User engagement → Interaction principle: The interactive system presents functions and information in an inviting and motivating manner supporting continued interaction with the system. Key questions: Does the solution motivate users to use it? Is the solution perceived as trustworthy by the users? Does the solution increase user involvement?

User evaluation: Reviews interface with users to ensure it is understandable, appropriate to tasks, and safe to use.

User expectations: The user's assumption or hope about what goal they will achieve using the system, product, or service.

User experience (UX): User's perceptions and responses that result from the use and/or anticipated use of a system, product, or service (ISO 9241-11:2018).

User Experience Questionnaire (UEQ): The UEQ is a questionnaire whose core was developed by Martin Schrepp in 2005. This questionnaire aims to measure the UX using a standardized procedure.

UX sprints: The usability and UX sprint is a smaller version of the usability and UX test. It is used when the execution of different iterations is planned directly, and the development cycle should be concise. Test execution and evaluation are usually shortened to one day to get results as quickly as possible.

User feedback: Reactions and feedback from users regarding the use of a product.

User group: *"Subset of intended users who are differentiated from other intended users by characteristics of the users, tasks, or environments that could influence usability"* (ISO 9241-11:2018).

User interface (UI): The UI is the interface between man and machine that allows the user to interact with the machine.

User interface (UI) design: The conceptual and visual design of the UI.

User interview → Interview: A UX research method. A guided conversation to evaluate user needs. Interviews can be structured, semi-structured, or open-ended.

User requirements: A prerequisite that must be given so that users can achieve their goals using a system, product, or service within a specific context of use.

User research: User data generation through focused observation, questioning, and listening.

User researcher: User researchers translate user needs into design requirements.

User stories: User stories are one of the most common ways to document requirements. A user story's structured formulation can describe a function's value for specific target groups.

User survey → Qualitative surveys → Guided interview: A (non-written) survey is a qualitative empirical method to elicit users' perceptions and opinions or user needs and requirements. The survey can be conducted with the help of a guideline but can also be semi-structured, consisting of a guideline and situational questions for more in-depth information. Both individual and group interviews are possible.

User tests: A group of methods to measure and improve the usability of a product. Users are observed covertly or overtly while using a product prototype to generate data on the product's usability.

User: The people who interact with a system, product, or service.

UX benchmarking: UX benchmarking is the quantitative assessment of the UX of a system, product, or service against metrics to measure those metrics against self-selected standards (e.g., previous versions, competing products, alternative designs), i.e., benchmarks.

UXer → Usability and UX designer: Usability and UX designers translate user requirements into design solutions.

Validity: Validity is a quality criterion for evaluating tests. Indicates whether the test measured what it claims to have measured.

Visual design: The visual design of an interface.

Web page slider → Slider: A slider is an element that displays information that changes in a certain time rhythm and can also be actively controlled by the user. Sliders are often used as a starting element on web pages to attract attention.

Web tracking → Click analysis: The tracking and evaluation of a user's click patterns when using a (web-based) software product to generate important usability data.

Wireframe: A wireframe is the basic structure of the screen that is represented schematically. Usually, a wireframe does not yet contain any visual design but is purely limited to representing the elements related to each other.

Workflow management system: A software or application that allows to display and manage workflows.

Workflow: A workflow is the sequence of work steps.

Appendix B
Bibliography

Adams, C. D. (1868). *The Genuine Works of Hippocrates*. Hippocrates. New York. Dover. Retrieved 2022 from *http://www.perseus.tufts.edu/hopper/text?doc=Perseus%3Atext%3A1999.01.0248%3Atext%3DOff.%3Asection%3D3*.

Adams, M. (2022). *Change Your Questions, Change Your Life*: *12 Powerful Tools for Leadership, Coaching, and Results*. 4th ed. National Geographic Books.

Agresti, A., Franklin, C., & Klingenberg, B. (2020). *Statistics: The Art and Science of Learning from Data*. Pearson.

American Psychological Association. (n.d.). Creativity. Retrieved May 6, 2023, from *https://www.apa.org/topics/creativity*.

American Psychological Association. (n.d.). Halo Effect. Dictionary of Psychology. Retrieved 2023 from *https://dictionary.apa.org/halo-effect*.

American Psychological Association. (n.d.). Rapport. Dictionary of Psychology. Retrieved May 6, 2023, from *https://dictionary.apa.org/rapport*.

American Psychological Association. (n.d.). Reactance Theory. Dictionary of Psychology. Retrieved May 6, 2023, from *https://dictionary.apa.org/reactance-theory*.

Anatomie Mensch. (2021). “Der Geruchssinn des Menschen - Die Nase.” *https://www.anatomie-mensch.com*.

Angrave, J. (2020). *The Journey Mapping Playbook*. Walter de Gruyter GmbH.

Armstrong, M. (2019). Global Data Creation Is About to Explode. Statista Daily Data. Retrieved 2023 from *https://www.statista.com /chart/17727/global-data-creation-forecasts/*.

Association, I. E. (2021). What is ergonomics? International Ergonomics Association. Retrieved from International Ergonomics Association: *https://iea.cc/what-is-ergonomics/*.

Atlassian Design Systems. (2023). Atlassian Guidelines. Retrieved from *https://atlassian.design/content/writing-guidelines/writing-a-warning-message*.

Austria-Forum. (2013). Bewusstseinsforschung. Retrieved 2023 from *https://austria-forum.org/af/Sparkling_Science/Aufsatzsammlung/Bewusstseinsforschung*.

Bangor, A., Kortum, P. & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies, 4*(3), 114–123.

Bartscher, T. (18.11.2021). Humanisierung der Arbeit. Gabler Wirtschaftslexikon. Retrieved from *https://wirtschaftslexikon.gabler.de/definition/humanisierung-der-arbeit-36234*.

Beasley, M. (2013). *Practical Web Analytics for User Experience: How Analytics Can Help You Understand Your Users*. Elsevier.

Bendel, O. (18.11.2021). Mensch-Maschine-Interaktion. Gabler Wirtschaftslexikon. Retrieved from *https://wirtschaftslexikon.gabler.de/definition/mensch-maschine-interaktion-54079*.

Berekoven, L., Eckert, W. & Ellenrieder, P. (2009). *Marktforschung: Methodische Grundlagen und praktische Anwendung*. Wiesbaden: Gabler.

Bojko, A. (2013). *Eye Tracking the User Experience. A Practical Guide to Research*. Rosenfeld Media.

British Design Council. (2007). A study of the design process – Eleven lessons: managing design in eleven global brands. London.

Brooke, J. (1996). SUS: A Quick and Dirty Usability Scale. In B. A. Weerdmeester, I. L. McClelland, & P. T. Jordan (Eds.), *Usability Evaluation in Industry* (pp. 189–194). Taylor & Francis.

Burmester, M., Hassenzahl, M. & Koller, F. (2007). Engineering attraktiver Produkte – AttrakDiff. In J. Ziegler & W. Beinhauer (Eds.), *Interaktion mit komplexen Informationsräumen* (pp. 121–141). Oldenbourg.

Carter, B. & Luke, S. (2020). Best Practices in Eye Tracking Research. *International Journal of Psychology*, 155, 49–62.

Christensen, C. M., Hall, T., Dillon, K. & Duncon, D. S. (2017). *Besser als der Zufall – »Jobs to Be Done« – die Strategie für erfolgreiche Innovationen*. Börsenmedien AG.

Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley.

Dark Horse Innovation. (2017). *Digital Innovation Playbook: The Essential Exercise Book for Founders, Doers and Managers*. Murmann.

Deutsches Institut Fur Normung (DIN). (2018). *Ergonomie der Mensch-System-Interaktion* – Teil 11: Gebrauchstauglichkeit: Begriffe und Konzepte. DIN.

Diefenbach, S., Lenz, E. & Hassenzahl, M. (2014). Experience Design Tools - Ansätze zur Interaktionsgestaltung aus dem Blickwinkel psychologischer Bedürfnisse. Retrieved from *https://dl.gi.de/bitstream/handle/20.500.12116/5760/Diefenbach_Lenz_Hessenzahl_2014.pdf?sequence=2#:~:text=Basierend%20auf%20einer%20Zusammenschau%20der,%2C%20Sicherheit%2C%20Selbstwert%20und%20Popularit%C3%A4t*.

Dilts, R. B. (1994). *Strategies of Genius*. Volume I: Aristotle, Sherlock Holmes, Walt Disney, Wolfgang Amadeus Mozart. Meta Publications.

Döring, N. & Bortz, J. (2016). *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften*. Springer.

Draxler, T. & Cheung, A. (2010). *30 Minuten Gesundheitsmanagement*. GABAL.

Dweck, C. S. (2006). *Mindset: The New Psychology of Success*. Random House.

Ebbinghaus, H. (1913). *On Memory: A Contribution to Experimental Psychology*. Teacher College.

Eppler, M. J., Hoffmann, F. & Pfister, R. A. (2017). *Creability: Gemeinsam kreativ – innovative Methoden für die Ideenentwicklung in Teams*. Schäffer-Poeschel Verlag für Wirtschaft Steuern Recht GmbH.

Ericsson, K., Krampe, R. & Tesch-Roemer, C. (1993). The Role of Deliberate Practice in the Acquisition of Expert Performance. *Psychological Review, 100*(3), 363–406.

Faulkner, L. (2003). Beyond the Five-User Assumption: Benefits of Increased Sample Sizes in Usability Testing. *Behavior Research Methods, Instruments, and Computers, 35*(3), 379–383.

Fitts, P. J. (1947). Psychological Aspects of Instrument Display. Analysis of 270 »pilot-error« experiences in reading and interpreting aircraft instruments (Report No. TSEAA-694-12A). Aero Medical Laboratory, Air Materiel Command, U.S. Air Force.

Fowler, T. (2022). Is it Legal to Drive with One Eye? Retrieved 2023 from *https://www.tomfowlerlaw.com/post/legal-to-drive-with-one-eye*.

Fürstenberg, F. (1981). *Gibt es eine »integrierte« Arbeitswissenschaft?* Europäisches Zentrum »Arbeit und Gesellschaft«.

Gesundheitsinformation. (2021). Wie funktioniert die Zunge?. Retrieved from *https://www.gesundheitsinformation.de*.

Gesundheitsinformation (18.11.2021). Aufbau der Zunge. Retrieved from *https://www.gesundheitsinformation.de/wie-funktioniert-die-zunge.html*.

Gerber, N. (2016). Let's get physical: Lässt sich User Experience physiologisch erfassen?. In *Mensch und Computer 2016 - Tagungsband*.

Gladwell, M. (2009). *Outliers: The Story of Success*. Penguin Group.

Goldstein, E. B., & Cacciamani, L. (2021). *Sensation and Perception* (Vol. 11).

Guilford, J. P. (1950). Creativity. *American Psychologist, 5*(9), 444–454.

Hague, P. (2021). *Market Research in Practice: An Introduction to Gaining Greater Market Insight*. Kogan Page.

Hamborg, K.-C. (18.11.2021). Mensch-Computer-Interaktion. Dorsch Lexikon der Psychologie. Retrieved from *https://dorsch.hogrefe.com/stichwort/mensch-computer-interaktion*.

Hassenzahl, M. (2010). *Experience Design: Technology for All the Right Reasons*. Springer Cham.

Hassenzahl, M. & Roto, V. (2007). Being and doing: A perspective on user experience and its measurement. *Interfaces* (72), 10–12.

Burmester, M., Hassenzahl, M., & Koller, F. (2002). Usability ist nicht alles - Wege zu attraktiven Produkten (Beyond Usability - Appeal of interactive Products). *i-com*.

Hassenzahl, M., Burmester, M., & Koller, F. (2003). AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In J. Ziegler, & G. Szwillus (Eds.), *Mensch & Computer 2003. Interaktion in Bewegung* (pp. 187–196). B.G. Teubner.

Hassenzahl, M., Burmester, M. & Koller, F. (2008). Der User Experience (UX) auf der Spur: Zum Einsatz von www.attrakdiff.de. In H. Brau, S. Diefenbach, M. Hassenzahl, F. Koller, M. Peissner, & K. Röse (Eds.), *Usability Professionals 2008* (pp. 78–82). German Chapter der Usability Professionals Association.

Hassler, M. (2012). *Web analytic*s. Verlagsgruppe Hüthig Jehle Rehm.

Hazenberg, W. (2012). *75 Tools for Creative Thinking: A Fun Card Deck for Everyone Who Needs Creative Inspiration*. BIS Publishers.

Heckhausen, J. & Dweck, C. S. (1998). *Motivation and Self-Regulation across the Life Span*. Cambridge University Press.

Heimann, M. & Schütz, M. (2017). *Wie Design wirkt – Psychologische Prinzipien erfolgreicher Gestaltung*. Rheinwerk Verlag.

Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H. & Van de Weijer, J. (2011). *Eye Tracking: A Comprehensive Guide to Methods and Measures*. Oxford University Press.

Hurtienne, J. & Prümper, J. (2003). Stress in the Office: the Influence of Software-Ergonomic Quality. In D. Harris, V. Duffy, M. Smith, & C. Stephanidis (Eds.), *Human-Centred Computing: Cognitive, Social, and Ergonomic Aspects* (pp. 63–67). Lawrence Erlbaum Associates.

IDG Research Services. (2018). Studie Arbeitsplatz der Zukunft 2018. Retrieved from *https://shop.computerwoche.de/fileserver/files/2002/412525429527272/leseprobe_arbeitsplatz_der_zukunft.pdf*.

Imai, M. (1986). *Kaizen: The Key to Japan's Competitive Success*. McGraw-Hill Publishing Company.

International Electrotechnical Commission (IEC). (2016). IEC TR 62366-2:2016 Medical devices - Part 2: Guidance on the application of usability engineering to medical devices. Retrieved 2023 from *https://www.vde-verlag.de/iec-normen/222801/iec-tr-62366-2-2016.html*.

International Ergonomics Association. (2021). What is ergonomics? Retrieved from *https://iea.cc/what-is-ergonomics/*.

International Organization for Standardization (ISO). (2020) ISO 9241-110:2020(en) Ergonomics of human-system interaction — Part 110: Interaction principles.

International Organization for Standardization (ISO). (2019). ISO/IEC. (12/2019). ISO 14971: Medical devices – Application of risk management to medical devices.

International Organization for Standardization (ISO). (2018). ISO 9241-11:2018(en) Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts.

International Organization for Standardization (ISO). (2016). ISO/TC 159/SC 4 Ergonomics of human-system interaction. ISO 9241-161:2016 Ergonomics of human-system interaction — Part 161: Guidance on visual user-interface elements.

International Organization for Standardization (ISO). (1998). ISO 9241-13:1998(en) Ergonomic requirements for office work with visual display terminals (VDTs) — Part 13: User guidance.

ISO/TC 159/SC 4 Ergonomics of human-system interaction. (2019) 24509:2019 Ergonomics - Accessible design - A method for estimating minimum legible font size for people at any age.

Kano, N., Seraku, N., Takahashi, F. & Tsuii, S. (1984). Attractive Quality and Must-Be Quality. *Journal of the Japanese Society for Quality Control, 14*(2), 39–48, 147–156.

Kirchgeorg, M. (2021). wirtschaftslexikon.gabler.de. Retrieved from *https://wirtschaftslexikon.gabler.de/definition/kunde-37108*.

Klatt, S. (07.06.2008). 20.000 Blitzentscheidungen pro Tag. Retrieved from *https://www.wiwo.de/5445178.html*.

Knemeyer, D. & Svoboda, E. (21.04.2022). The Glossary of Human Computer Interaction – User Experience – UX. Interaction-design.org. Retrieved from *https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/user-experience-ux*.

Knutson, B., Rick, S., Wimmer, E., Prelec, D. & Loewenstein, G. (January 2007). Neural Predictors of Purchases. *Neuron*, p. 147–156.

Kriesl, M. (2013). Bewusstseinsforschung. Austria-Forum.org. Retrieved 2023 from *https://austria-forum.org/af/Sparkling_Science/Aufsatzsammlung/Bewusstseinsforschung*.

Laricchia, F. (2022, October 18). Smartphone users in the US 2009-2040. Retrieved from *https://www.statista.com/statistics/201182/forecast-of-smartphone-users-in-the-us/*.

Laugwitz, B. & Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In A. Holzinger (Eds.,), USAB 2008, LNCS 5298, S. 63–76.

Laugwitz, B., Schrepp, M. & Held, T. (2008). Construction and evaluation of a user experience questionnaire. HCI and Usability for Education and Work, 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Lecture Notes in Computer Science, Vol 5298, p. 63–76. Graz.

Lewrick, M., Link, P. & Leifer, L. (2020). *The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods*. John Wiley & Sons.

Lewrick, M., Link, P. & Leifer, L. (2018). *The Design Thinking Playbook: Mindful Digital Transformation of Teams, Products, Services, Businesses and Ecosystems*. John Wiley & Sons.

Liggieri, K. & Müller, O. (2019). *Mensch-Maschine-Interaktion – Handbuch zu Geschichte – Kultur — Ethik*. Springer-Verlag GmbH.

Lindgaard, G., Fernandes, G., Dudek, C. & Brown, J. (März 2006). Attention Web Designers: You Have 50 Milliseconds to Make a Good First Impression! *Behaviour & Informations Technology, 25*, 115–126.

Lindsay, P. H., & Norman, D. A. (1977). Human Information Processing: An Introduction to Psychology. Academic Press.

Luczak, H. (1975). Untersuchungen informatorischer Belastung und Beanspruchung des Menschen. *Fortschrittsberichte der VDI-Zeitschriften, 10*(2). 13.

Luczak, H. & Volpert, W. (1987). *Arbeitswissenschaft. Kerndefinition – Gegenstandskatalog – Forschungsgebiete*. RKW-Verlag.

Macnamara, B. N. & Maitra, M. (2019). The Role of Deliberate Practice in Expert Performance: Revisiting Ericsson, Krampe, & Tesch-Römer (1993). *Royal Society Open Science, 6*, 1–19.

Mahto, D. & Kumar, A. (2008). Application of Root Cause Analysis in Improvement of Product Quality and Productivity. *Journal of Industrial Engineering and Management, 1*(2), 16–53.

Majid, A., Roberts, S. T., Cilissen, L., Emmorey, K., Nicodemus, B., O'Grady, L., Woll, B., Lelan, B., De Sousa, H., Cansler, B. L., Shayan, S., De Vos, C., Senft, G., Enfield, N. J., Razak, R. A., Fedden, S., Tufvesson, S., Dingemanse, M., Ozturk, O., Levinson, S. C. (2018). Differential coding of perception in the world's languages. *Proceedings of the National Academy of Sciences*, *115*(45), 11369–11376. *https://doi.org/10.1073/pnas.1720419115*.

Marcano, A. (03/2011). Retrieved from *http://antonymarcano.com/blog/2011/03/fi_stories/*.

Maslow, A. (1954). The Instinctoid Nature of Basic Needs. *Journal of Personality*. 326-347.

Mayer, D., Schrepp, M. & Held, T. (2018). Beurteilung der UX Qualität durch Experten. In *Mensch und Computer 2018 – Usability Professionals* (pp. 3–14). Gesellschaft für Informatik e.V. und German UPA e.V.

Mayring, P. (2002). Qualitative Content Analysis – Research Instrument or Mode of Interpretation? In M. Kiegelmann (Ed.), *The Role of the Researcher in Qualitative Psychology* (pp. 139–148). Huber.

Miles, M. B., Huberman, A. M., & Saldana, J. (2019). *Qualitative Data Analysis: A Methods Sourcebook*. SAGE Publications.

Merriam-Webster Dictionary. (2023). Analysis. Retrieved 2023 from *https://www.merriam-webster.com/dictionary/analysis*.

Michalko, M. (2006). *Thinkertoys: A Handbook of Creative-Thinking Techniques*. Ten Speed Press.

Mindspace. (2018). marktforschung.de: Kündigungsgrund Büroausstattung. Retrieved from *https://www.marktforschung.de/aktuelles/meinung/marktforschung/kuendigungsgrund-bueroausstattung/*.

Moran, K. (2020). Benchmarking UX: Tracking Metrics. Nielsen Norman Group. Retrieved from *https://www.nngroup.com/articles/benchmarking-ux/*.

Müller, J. (10.02.2010). Das Geheimnis attraktiver Produkte – und wie man Attraktivität messen kann. Retrieved from *https://www.usabilityblog.de/das-geheimnis-attraktiver-produkte-und-wie-man-attraktivitat-messen-kann/#:~:text=Die%20wahrgenommene%20hedonische%20Qualit%C3%A4t%20erf%C3%BCllt,%2C%20beeindruckend%2C%20aufregend%20und%20interessant*.

NAErg, D.-N. E. (2019). Ergonomie der Mensch-System-Interaktion – Teil 210: Menschzentrierte Gestaltung interaktiver Systeme (ISO 9241-210:2019). Berlin.

Natoli, J. (2017). How to Write Smarter User Stories for Product Design and Development. Retrieved from *https://www.uxpin.com/studio/blog/write-smarter-user-stories-product-design-development/*.

Nielsen, J. & Pernice, K. (2010). *Eyetracking Web Usability*. New Riders.

Nielsen, J., Berger, J.M., Gilutz, S. & Whitenton, K. (n.d.) Return on Investment (ROI) for Usability. 4th edition.

Occupational Safety and Health Administration. (n.d.). Occupational Noise Exposure. Retrieved 2023 from *https://www.osha.gov/noise*.

Ohly, S., Sonnentag, S., Niessen, C. & Zapf, D. (2010). Diary Studies in Organizational Research: An Introduction and Some Practical Recommendations. *Journal of Personnel Psychology*, 9(2).

Osterwalder, A., Pigneur, Y., Etiemble, F. & Smith, A. (2020). *The Invincible Company*. Campus Verlag GmbH.

Partala, T., Surakka, V. & Vanhala, T. (2006). Real-Time Estimation of Emotional Experiences from Facial Expressions. *Interacting with Computers, 18*, 208–226.

Patten, M. L., & Newhart, M. (2017). *Understanding Research Methods: An Overview of the Essentials*. Routledge.

Penfield, W. & Rasmussen, T. (1950). *The Cerebral Cortex of Man: A Clinical Study of Localization of Function*. Macmillan.

Podmajersky, T. (2019). *Strategic Writing for UX: Drive Engagement, Conversion, and Retention with Every Word*. O'Reilly Media.

Rabinowitz, T. (2015). *Exploring Typography*. Cengage Learning.

Renner, K. & Jacob, N. C. (2020). *Das Interview: Grundlagen und Anwendung in Psychologie und Sozialwissenschaften*. Springer.

Rhodes, M. (1961). An Analysis of Creativity. *The Phi Delta Kappan, 42*(7), 305–310.

Ries, E. (2011). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business.

Rittel, H. W. & Webber, M. M. (1973). Dilemmas in a General Theory of Planning. *Policy Sciences, 4*(2), 155–169.

Ritter, S. M. & Ferguson, S. (2017). Happy Creativity: Listening to Happy Music Facilitates Divergent Thinking. *PLOS ONE*.

SAE International. (2018). Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. *On-Road Automated Driving (ORAD) Committee*. *https://doi.org/10.4271/j3016_201806*.

Sahakian, B. & LaBuzetta, J. N. (2013). *Bad Moves: How Decision Making Goes Wrong, and the Ethics of Smart Drugs*. Oxford University Press.

Sarodnick, F. & Brau, H. (2006). *Methoden der Usability Evaluation*. Verlag Hans Huber.

Schlick, C., Bruder, R. & Luczak, H. (2010). *Arbeitswissenschaft*. Springer-Verlag.

Schrepp, M. (2019). User Experience mit Fragebögen messen. Mensch und Computer 2019 – Usability Professionals. Gesellschaft für Informatik e.V. und German UPA e.V.

Schrepp, M., Hinderks, A. & Thomaschewski, J. (2014). Applying the User Experience Questionnaire (UEQ) in Different Evaluation Scenarios. In A. Marcus (Ed.), *Lecture Notes in Computer Sciences*, 8517, pp. 383–392.

Schrepp, M., Hinderks, A. & Thomaschewski, J. (2017a). Construction of a Benchmark for the User Experience Questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence, 4*(4), 40–44.

Schrepp, M., Hinderks, A. & Thomaschewski, J. (2017b). Design and Evaluation of a Short Version of the User Experience Questionannire (UEQ-S). *International Journal of Interactive Multimedia and Artificial Intelligence, 4*(6), 103–108.

Schmidt-Thieme, L. & Gaul, W. (2002). Aufzeichnung des Nutzerverhaltens—Erhebungstechniken und Datenformate. In Hippner, H., Merzenich, M., Wilde, K.D. (Ed). *Handbuch Web Mining im Marketing*, 35–53.

Sheldon, K. M., Elliot, A. J., Kim, Y. & Kasser, T. (2001). What Is Satisfying about Satisfying Events? Testing 10 Candidate Psychological Needs. *Journal of Personality and Social Psychology, 80*(2), 325–339. doi:10.1037//0022-3514.80.2.325.

Siepermann, M. (18.11.2021). Software-Ergonomie: Gabler Wirtschaftslexikon. Retrieved from *https://wirtschaftslexikon.gabler.de/definition/software-ergonomie-44137*.

Siroker, D. & Koomen, P. (2013). *A/B-Testing: The Most Powerful Way to Turn Clicks into Customers*. Wiley.

Spencer, D. (2009). *Card Sorting: Designing Usable Categories*. Rosenfeld Media.

Stangl, W. (09.09.2021). Kreativität – Online Lexikon für Psychologie und Pädagogik. Retrieved from *https://lexikon.stangl.eu/542/kreativitaet*.

Starrett, K. & Cordoza, G. (2016). *Sitzen ist das neue Rauchen: Das Trainingsprogramm, um lebensstilbedingten Haltungsschäden vorzubeugen und unsere natürliche Mobilität zurückzugewinnen*. Münchner Verlagsgruppe GmbH.

Stegemann, M. & Suwelack, T. (2020). A/B-Testing – Verfahren zur Optimierung der digitalen Interaktion zwischen Konsumenten und Unternehmen. In S. Boßow-Thies, Hofmann-Stölting, C. & H. Jochims (Eds.), *Data-Driven Marketing*. Springer Gabler.

Stoltzfus, T. (2008). *Coaching Questions: A Coach's Guide to Powerful Asking Skills*. Pegasus Creative Arts.

Szabo, P. W. (2017). *User Experience Mapping*. Packt.

Tambini, A., Rimmele, U., Phelps, E. et al. (2017). Emotional Brain States Carry Over and Enhance Future Memory Formation. *Nature Neuroscience, 20*, 271–278. *https://doi.org/10.1038/nn.4468*.

Taron, K., Pfeiffer, T. & Thomaschewski, J. (2016). *UX schnell und effektiv mittels 5-Sekunden-Test*. UP 2016.

Tedesco, D. & Tranquada, F. (2013). *The Moderator's Survival Guide: Handling Common, Tricky, and Sticky Situations in User Research*. Elsevier.

Tonhauser, P. (2018). *Design Thinking Workshop – 12 Komponenten, die in keinem Design Thinking Workshop fehlen dürfen*. eBook.

Tossell, C., Finomore, V., Endsley, M., Wickens, C., Bennett, W., Knott, B. & McClernon, C. (2016). Human Factors and the United States Military: A 75-Year Partnership. Proceedings of the Human Factors and Ergonomics Society Annual Meeting (pp. 91–93). doi:10.1177/1541931213601020.

Ullrich, D. (2013). Komponenten und Einflussfaktoren der intuitiven Interaktion: Ein integratives Modell. i-com. *Zeitschrift für interaktive und kooperative Medien, 12*(3), 44–53.

Ullrich, D. (2014). *Intuitive Interaktion: Eine Exploration von Komponenten, Einflussfaktoren und Gestaltungsansätzen aus der Perspektive des Nutzererlebens*. Doktorarbeit. Darmstadt.

Ullrich, D. & Diefenbach, S. (2010). INTUI. Exploring the Facets of Intuitive Interaction. *Mensch & Computer* 2010 (pp. 251–260). Oldenbourg.

Voigt, K.-I. (18.11.2021). Rationalisierung: Gabler Wirtschaftslexikon. Retrieved from *https://wirtschaftslexikon.gabler.de/definition/rationalisierung-41970*.

Voss, C., & Raz, T. (2017). *Never Split the Difference: Negotiating as If Your Life Depended on It*. Century.

Wäger, M. (2017). *Das ABC der Farbe: Theorie und Praxis für Grafiker und Fotografen*. Rheinwerk Verlag.

Weber, E. H. (1835). Über den Tastsinn. *Archiv für Anatomie, Physiologie und wissenschaftliche Medizin. https://vlp.mpiwg-berlin.mpg.de/library/data/lit29424?*, 152-159.

Wertheimer, M. (1923). *Untersuchungen zur Lehre von der Gestalt*. Julius Springer.

Wiklund, M., Dwyer, A. & Davis, E. (2016). *Root Cause Analysis – Medical Device Use Error*. CRC Press.

Witt, J. & Witt, T. (2006). Der kontinuierliche Verbesserungsprozess. (KVP). Konzept – System – Maßnahme (= Arbeitshefte Führungspsychologie. Band 42). 2., durchgesehene Auflage. Verlag Recht und Wirtschaft.

Wood, J. & Wood, L. (2008). Card Sorting: Current Practices and Beyond. *Journal of Usability Studies, 4*(1), 1–6.

Wübbenhorst, K. (09.09.2021). Objektivität – Gabler Wirtschaftslexikon. Retrieved from *https://wirtschaftslexikon.gabler.de/definition/objektivitaet-45537*.

Wübbenhorst, K. (09.09.2021). Reliabilität – Gabler Wirtschaftslexikon. Retrieved from *https://wirtschaftslexikon.gabler.de/definition/reliabilitaet-44718*.

Wübbenhorst, K. (09.09.2021). Validität – Gabler Wirtschaftslexikon. Retrieved from *https://wirtschaftslexikon.gabler.de/definition/validitaet-49676*.

Yifrah, K. (2019). *UX Writing & Microcopy*. Rheinwerk Verlag.

Appendix C
The Authors

Benjamin Franz received his doctorate in engineering; his UX-related dissertation on highly automated driving was awarded the Walter Rohmert Research Prize. In his jointly founded and managed company, Custom Interactions, he focuses on user interface design. He also works as a lecturer and a keynote speaker.

Michaela Kauer-Franz has a doctorate in psychology with a focus on product design and user experience. Together with Benjamin Franz, she founded the data-driven UX agency Custom Interactions, which they manage together. She also contributes her extensive experience as a trainer, a lecturer at Technische Universität Darmstadt, a speaker, and on an International Organization for Standardization (ISO) committee.

Index

2-point idea ... 432
2-point threshold ... 214
3-point idea ... 432
5-second test ... 557
  profile ... 560

A

A/B testing ... 368, 548, 625
  profile ... 552
Abstraction ... 379
Accommodation ... 191, 196, 202
Accordion ... 457
Action problem ... 609
Active listening ... 179, 347
Actual operating time ... 535
Additive color mixture ... 256
Address data ... 453
Adobe XD ... 440, 490
Affective elements ... 298
Agile development ... 314
Analysis ... 322, 573
  advertising ... 559
  context of use ... 49
  customer needs ... 111
  eye movements ... 594
  facial expressions ... 42
Android Material Design ... 465
Answer options ... 449
Areas of interest ... 595
Asking questions ... 175
  closed questions ... 177
  open-ended questions ... 175
Assessments ... 630
Associations ... 190
  colors ... 248
  primary and secondary ... 592
  surveys ... 247
Association space ... 244, 246, 500
Association space test (ARTTest) ... 247, 591
  profile ... 594
Assumptions ... 150, 151, 164, 322
  check ... 91, 182
  confirm ... 91
  revealing ... 150
  typically unchecked ... 157
  versus knowledge ... 151, 154, 157, 309
Assumptions-knowledge matrix ... 151, 309
Attention ... 226
AttrakDiff ... 566, 593
Axure RP ... 440, 488

B

Balsamiq ... 440
Behavioral design ... 245
Benchmarking ... 536, 552
  profile ... 557
Beta features ... 317
Biased data ... 523
Binocular depth cues ... 200
Binocular disparity ... 200
Braille ... 213
Brainstorming ... 92, 93, 95, 429
Brainwriting ... 94, 95, 429, 509
Breadcrumbs ... 281
British Design Council ... 104
Build-measure-learn cycle ... 519
BusinessBooster ... 30, 58, 169, 247, 266, 339
  context analysis ... 393
  design steps ... 511
  evaluation ... 612
  measure ROI ... 77
  requirements ... 421
  what's next ... 623
Business success ... 64, 71
Business-to-business (B2B) ... 24, 123, 360
  BusinessBooster ... 30
  customers ... 47
  Healthyfy ... 32
  metrics ... 71
Business-to-consumer (B2C) ... 24, 71, 360
  customers ... 47
  Healthyfy ... 32
  metrics ... 71
Button ... 445
  information button ... 468
  radio button ... 449
  toggle button ... 458
Buyer ... 46

C

Calendar ... 461
Call-to-action (CTA) ... 250

Card sorting ... 474, 582
  profile ... 586
Carousel ... 474
Categorization ... 145
Checkbox ... 455
Chunks ... 239
Ciliary muscle ... 191
Circles ... 260
Circumvallate papillae ... 219
Classic requirements ... 411
Click analysis ... 366, 576
  profile ... 369
Click dummy ... 75, 485, 487, 611
  testability ... 491
  tools ... 488
Cochlea ... 205
Cochlear partition ... 205
Cognitive task ... 328
Color coding ... 198, 254
Colorizer ... 251
Color palette ... 467
Color picker ... 466
Colors ... 248
Color space ... 256
  CYMK ... 257
  RGB ... 256
Color vision ... 193
  deficiency ... 196, 197, 256
Combo box ... 447, 454
Competitive solutions ... 427
Complementary colors ... 251
Comprehensibility ... 524
Concept engineer ... 444
Conditioning ... 241
Cones ... 194
Confidentiality ... 523
Confirmation dialog ... 470
Conformity with user expectations ... 284
Consistency (internal and external) ... 285
Constructivism ... 242
Consumer Reports ... 517
Content consideration ... 442
Context of use ... 41, 84, 321, 627, 631
  analysis ... 122, 153, 322, 333, 519, 580
  analytical methods ... 373
  components ... 324
  contextual inquiry ... 360
  definition ... 49, 321
  description ... 85
  practical examples ... 391
  usage requirements ... 124
  user requirements ... 401
Contextual inquiry ... 360
  profile ... 363
Contextual interview ... 360
Contrast ... 254
Controllability ... 290
Controls to navigate ... 282
Convergence ... 199
Convergent thinking ... 104
Conversion rate ... 74, 366, 550
Corpus size ... 264
Cover flow ... 474
Covert observation ... 171
Creativity ... 183
Criticality ... 254
  workflow ... 318
Cross ... 260
Customer ... 46, 380
Customer experience ... 47
Customer journey mapping ... 380
  profile ... 384
Customer lifetime value ... 66
Customizability ... 293

D

Dark mode ... 201
Data
  availability ... 365
  evaluation ... 141
  interpretation ... 142
  meaningful ... 144
  objective ... 137, 142
  qualitative ... 140
  quantitative ... 140
  subjective ... 137
Data-driven UX design ... 49, 82, 117, 602, 617, 635
  choose the right approach ... 622
  context of use analysis ... 324
  design ... 494
  developing design solutions ... 425
  effort ... 317
  evaluation ... 515
  goal ... 117
  in agile projects ... 314
  phases ... 617
  planning ... 305, 306
  process ... 119
  secondary research ... 334
  user requirements ... 401
  working with prototypes ... 433

Date picker ... 464
Date selection ... 463
Decision-maker veto ... 432
Decision matrix for method selection ... 625
Decision problem ... 609
Default ... 448
Depth effect ... 258
Depth perception ... 199, 203
Derive requirements ... 85
Dermis ... 212
Design ... 43
  basic visual elements ... 247
Design compass ... 504
Design expertise ... 164, 167
Design grid ... 481, 496
Design impact ... 630
Design research ... 426, 429
  visual ... 497
Design solutions ... 86, 425
  develop ... 126
  evaluate ... 128
Design thinking ... 88, 389, 430
  define point of view ... 91
  ideate ... 92
  observe ... 91
  prototype ... 98
  test ... 99
Desk research ... 82
Desktop versus mobile ... 496
Detection problem ... 609
Detection threshold ... 222
Deuteranopia ... 197
Dial ... 460
Dialog ... 446
Diary study ... 577
  profile ... 582
Digital product design ... 425
Direct observation ... 171
Discover ... 105
  controls ... 287
  requirements ... 121
Display format ... 479
Divergent thinking ... 104
Documentation ... 312, 429, 588, 610
Dots ... 260
Double Diamond ... 104
  define ... 106
  deliver ... 107
  develop ... 106
  discover ... 105
Double opt-in ... 419
Dropdown ... 447, 465

E

Ear ... 203
Eardrum ... 204
Echoic memory ... 238
Effectiveness ... 41, 51, 58–60
Efficiency ... 41, 51, 58, 59
Emotional level ... 329
Empathy map ... 384, 388, 402
  profile ... 391
Environment ... 331
Epidermis ... 212
Episodic memory ... 241
Ergonomics ... 36, 38, 50, 56, 58, 60
  human-system-interaction ... 278
Evaluation ... 87, 128, 515
  by experts ... 521
  by users ... 521
  ideas ... 431
  matrix ... 95, 567
  medical devices ... 610
  methods ... 529
  table ... 348
Excitement attribute ... 55
Expectation management ... 308, 311
Experience ... 183
Expert evaluation ... 521
Expertise ... 164
Expert review ... 587
  profile ... 590
Explicit memory ... 241
Extended Color Gamut (ECG) ... 257
External auditory canal ... 204
Eye ... 191
Eyedropper ... 467
Eye tracking ... 594, 599
  profile ... 599

F

Familiar size ... 199
FAQ section ... 457
Feature requests ... 67, 370, 573
Feedback ... 113, 301, 317, 519
  analysis ... 369
  objective versus subjective ... 608
Field observation ... 339, 341, 392
  profile ... 342
Figma ... 440, 490

Filiform papillae ... 219
First impression ... 232, 629
First-mover advantage ... 386
Fixation ... 191, 193
Fixed mindset ... 149, 186
Fixed requirements ... 411
Flexibility ... 131, 621
Flexibility and risk matrix ... 130, 135, 142, 316, 620
Fly-on-the-wall method ... 340
Focus group ... 351, 352, 397
  profile ... 354
Foliate papillae ... 219
Follow-up surveys ... 534
Font size ... 264
Font type ... 265
Formative evaluation ... 515, 519, 520
Formative testing ... 515
Fovea centralis ... 192
Frame of reference ... 143
Free evaluation ... 588
Free response ... 357
Frequency ... 333
Frequency of use ... 128, 364, 573, 579, 631
Friendly user ... 162
Functional level ... 329
Functional scope ... 75, 101
Fungiform papillae ... 219

G

Gain point ... 385, 389
Gestalt principles ... 269, 277
Glomerulus ... 222
Goal ... 327, 376
  business success ... 64
  design ... 425
  levels ... 328
  user ... 322
Goal expertise ... 164, 165
Google Search ... 454
Graphics ... 266
Group discussions ... 350
  profile ... 354
Growth mindset ... 149, 186
Gut feeling ... 569

H

Hair cells ... 206
Halo effect ... 162
Hard facts ... 166
Hardware ... 330
Healthyfy ... 32, 60, 169, 243, 266, 339
  context analysis ... 397
  design steps ... 512
  evaluation ... 614
  measure ROI ... 78
  requirements ... 422
  visual design ... 506
  what's next ... 624
  wireframe ... 439
Hearing ... 207
Heat map ... 595
Hedonic quality ... 562
Heuristics ... 588
Hierarchy of needs ... 165
Highest paid person's opinion (HIPPO) ... 119, 307
High-fidelity prototype ... 436, 440
Horizontal prototype ... 434
How-might-we question ... 92, 308
Human-centered design ... 83
Human-computer interaction ... 40
Human factors ... 38, 82
Human Interface Guidelines ... 441
Human-machine interaction ... 39

I

Icon ... 268, 478
IEC 62366-1 (Standard) ... 108
Images ... 266, 478
Immersion ... 216
Implicit memory ... 241
Increments ... 314
Incus ... 205
Independent testing institutions ... 517
Indirect observation ... 171
Influencers ... 518
Information ... 331
Information processing ... 225
  stimulus-response-model ... 225
Input field ... 451
Interaction principles ... 278, 302
Interactive prototype ... 487
Interdisciplinary ... 634
Interference in the perception ... 254
Internal communication ... 157
Interval requirements ... 413
Interview ... 175, 180, 343, 625
  evaluation table ... 348
  guide ... 178, 344
  observations versus ... 177

Interview (Cont.)
  on-site 392
  possible questions 345
  profile 350
  qualitative 343
  semistructured 178
INTUI 568

J

Jobs theory 166, 328
Joy of use 43

K

Kano model 54
Key performance indicator (KPI) 364, 618
Kick-off workshop 158
Knowledge 150, 183, 334
  about processes 628
  gaps 519
  users 627
  versus assumptions 151, 154, 157, 309

L

Label 478
Law of closure 271
Law of common fate 276
Law of continuity 273
Law of figure/ground 275
Law of proximity 269
Law of similarity 270
Law of symmetry 274
Law of the point of view 274
Law of visual hierarchy 277
L-cones 194
Learnability 287
Learning 186
  design 183
Legal framework 574
Legibility 263
Lens 191
Light-dark adaptation 196
Light receptors 194
Light sensitivity 196
Light stimuli 191
Likert scale 355
Limbic system 240
Lines 261
Listening 179
  active 179
Low-code 508
Low-fidelity prototype 435

M

Macrosmatics 220
Macula 191
Magical experience 569
Malleus 205
Maluma 243
Material 262, 330
Material Design Guidelines 442
MAUDE database 372
Maximum requirements 415, 416
Measurable value 418
Mechanoreceptors 212
Medical devices 108, 325, 610
  evaluation 610
  requirements 610
Memory 229, 237
  chunks 239
  echoic memory 238
  episodic 241
  explicit 241
  implicit 241
  rehearsal 239
  semantic 241
  sensory register 238
  short-term memory 238
  ultra-short-term memory 238
Menu bar 472
Method selection 625
Metrics 65
  B2B versus B2C 71
  customer-related 66
  selection 68
  service-related 67
  user-related 66
Microsmatics 220
Microsoft PowerPoint 440
Middle ear 205
Mind map 336
Mindset 149, 181
  fixed mindset 149, 186
  growth mindset 149, 186
  scientific 156, 181
Minimum requirements 414
Minimum viable product (MVP) 100, 317, 519
  build 102
  learn 103
  measure 102

Mobile first 492, 493
Mobile systems 595
Monocular depth cues 199
Mood board 503–506
Motion-induced depth cues 199
Motion parallax 200
moveHUB 29, 56, 168, 172, 339
  context analysis 391
  design steps 509
  evaluation 611
  measure ROI 76
  requirements 420
  user versus customer 48
  what's next 623
Multiple-choice questions 356

N

Net Promoter Score (NPS) 66
Neutral display 448
Nociceptors 212, 217
No-code 508
Nose 220

O

Objective data 137, 142
Objective feedback 608
Observations 171
  advantages and limitations 174
  documentation 172
  field observation 339
  good 174
  outline 173
  types 171
  versus interviews 177
Observe 91
Odor profile 222
Olfactory mucosa 220
Onboarding 577
Open-ended questions 175
Open observation 171
Operating error 108, 430
Opportunity costs 410
Organizational fine-tuning 311
Organ of equilibrium 206
Outer ear 204
Overlay 471

P

Pain point 385, 389
Panels 358
Password field 469
Patient 108, 151, 322
  medical devices 325
Peer-reviewed journals 338
Perception 190, 634
  images versus text 266
  influencing factors 247
Perceptual capability (absolute and relative) 197, 210
Perceptual task 328
Performance attribute 55
Persona 359, 384, 392, 395, 398, 402, 420
  creation 386
  profile 390
Physical environment 332
Physical product design 425
Physical reactions 599
Physiological measurements 599
  profile 602
Pivot 623
Placeholders 478
Planning 305
Plausibility check 358
Point of no return 306
Popups 469
Postcondition 529
Pragmatic quality 562
Precondition 526
Primacy-recency effect 156
Priming 242
Prioritization 315
Problem-solution fit 535
Problem space 89
Procedural knowledge 241
Productivity 73
Proof of concept 32
Protanopia 197
Proto-mood board 503–505
Proto-persona 387
Prototype 98, 126, 433, 534, 536
  click dummy 487
  decision matrix 438
  high-fidelity 436, 440
  horizontal versus vertical 434
  low-fidelity 435
  testing 99, 548, 558, 578, 591, 597
Psychology 189
Pupil 191

Q

Qualitative data 140, 142
Quality 481, 618, 622
Quality gate 619
Quantitative data 140
Quantitative survey 355
Questionnaire 144, 355, 616
  design 145
  profile 359, 572
  standardized 561

R

Range requirements 412
Rapid user test 544
  profile 547
Rapport 176, 180, 182
Reactance 175
Recognition problem 609
Recognition threshold 222
Rectangles 259
Reflective design 245
Regulatory environment 332
Rejection attribute 55
Relative size 199
Remote testing 522
Requirements 72, 85, 124, 401
  absence of contradictions 86
  classic 411
  collect 85, 87, 95
  documentation 404, 416
  fixed 411
  general 626
  iteration 634
  list 85, 416, 421
  maximum 415
  minimum 414
  range 412
  specifications 416, 421
Resonance range 204
Resources 329
Responsive design 494, 495
Retina 191
Return on investment (ROI) 64, 75, 364, 618
  plan instead of discover 69
  studies 73
Risk 71, 133, 316, 621
  assessment 117
  classification 123
  consideration 333
  design 127

Risk (Cont.)
  effort 318
  evaluation 128
Rods 194
Rolling wave planning 313
Root cause analysis 602, 603

S

Saccule 207
Sample size 532
Sans-serif fonts 265
Satisfaction 41, 42
Scale ratings 141
Schedule 351
S-cones 194
Score poll 432
Screener 347, 531, 553
Screen flow 416, 436, 485, 608
Screen recording 545
Screen sizes 493
Scrum method 313
Search fields 453
Secondary research 82, 334, 338
  profile 338
Self-descriptiveness 281
Semantic differential 355, 565
Semantic memory 241
Semi-structured planning 305
Sense of touch 213
Sensory cells 212
Sensory organs 190
  ear 203
  eye 191
  nose 220
  skin 211
  tongue 218
Sensory register 238
Serifs 265
Shadowing 339, 340, 392
Shadows 199
Shape coding 198
Shapes 258
Short-term memory 238
Simulation 86, 289
Simultaneous sense 193
Skin 211
Slider 460, 477
Smartphones 493
Smartwatches 493
Social level 329
Software 330

Solution neutral ... 403
Solution space ... 89
Sound pressure level ... 207
Specifications ... 401
Speed ... 113
Sprint ... 313, 540
  cycles ... 540
  second ... 315
Squares ... 259
Stakeholder ... 72, 501
  workshop ... 306, 308, 504
Standardized test procedures ... 553
Standard symbols ... 230
Stapes ... 205
State-of-the-art design ... 426
Stationary solutions ... 594
Stimulus-response model ... 225, 603
  act ... 235
  decide ... 232
  discover ... 226
  memory ... 237
  reaction ... 237
  recognize ... 229
  stimulus ... 226
Style guide ... 75, 412, 416, 507
Subcutis ... 212
Subjective data ... 137
Subjective feedback ... 608
Subtractive color mixing ... 257
Successive sense ... 203
Suitability for the user's task ... 279
Summative evaluation ... 130, 515, 519
Summative testing ... 515
Supported evaluation ... 588
Support ticket ... 370, 573, 575, 635
Survey ... 355
  follow-up surveys ... 534
  online ... 547
  period ... 581
  profile ... 359
Survey guide ... 580
Swimlanes ... 380
System Usability Scale (SUS) ... 562

T

Takete ... 243
Talent ... 183
Target groups ... 405
Task ... 327, 376
  abstracted version ... 377
Task analysis ... 341, 373
  profile ... 379
Task expertise ... 164, 165
Taste perception ... 219
Taste receptors ... 218
Test case ... 525
Test guide ... 532, 537, 542, 553, 555, 558, 559, 562
Testing ... 515, 629
  company culture ... 186
  medical devices ... 610
  methods ... 529
Text block ... 478
Text field ... 451
Texture ... 262
Texture gradient ... 199
Threshold attribute ... 55
Ticket analysis ... 370
  profile ... 373
Time approach ... 619, 622
Time picker ... 464
Tongue ... 218
Tool tip ... 468
Touch-based operation ... 474
Touchpoint analysis ... 380, 383
  profile ... 384
Trends ... 499
Triangles ... 260
Tritanopia ... 197
Trustworthiness ... 300
Typography ... 263, 501

U

Ultra-short-term memory ... 238
Undo function ... 293
Unified Modeling Language (UML)
  flowchart ... 375
Usability ... 24, 35, 41, 45, 50, 72, 161, 535, 556, 593
  assess ... 566, 588, 590
  definition ... 41
  issue ... 364
  medical devices ... 108
  sprint ... 540, 544
  test ... 530, 539, 612, 629
  usability engineering ... 108
Usage behavior ... 630
Usage data ... 364, 368, 372, 573
  patterns ... 365
  profile ... 366, 575
Usage environment ... 326, 526, 627

Use case ... 404, 494, 541
Use error ... 295
Use error robustness ... 294
User ... 63, 633
  context of use analysis ... 322
  definition ... 44, 71, 324
  engagement ... 298
  evaluation ... 521
  feedback ... 519
  feedback data ... 371
  flows ... 485
  groups ... 45, 553, 626
  instructions ... 527
  journey ... 443
  journey mapping ... 380
  participation ... 301
  requirements ... 82
User expectations ... 42
User experience (UX) ... 24, 35, 50
  benchmarking ... 536, 552, 557
  metrics ... 66
  rapid user test ... 544
  sprint ... 540, 544
  test ... 530, 539, 612, 629
User Experience Questionnaire (UEQ) ... 564
User interface (UI) ... 39, 248, 430
  affective elements ... 298
  interaction principles ... 278
  requirements ... 416
  switch versions ... 301
  visual design ... 497
User manual ... 159
User needs ... 42
User research ... 44, 74, 91, 170, 315, 325, 425, 515
User researcher ... 170, 181
User stories ... 405
  conventional ... 406
  extensions ... 408
  reformulated ... 407
User tests ... 541
  assumptions ... 163
  Double Diamond model ... 107
User tests (Cont.)
  evaluation of design solutions ... 130
  with prototypes ... 99
Utricle ... 206
UXer ... 617, 634

V

Validity ... 572
Value range ... 460
Verbalizability ... 569
Vertical guide ... 482
Vertical prototype ... 434
Vestibular perception ... 206
Vibration ... 215
Visceral design ... 244
Visual design ... 430, 497
  concept ... 506
Visualization ... 114

W

Warning ... 201, 480
  symbols ... 412
  tones ... 210
Waterfall model ... 314
Web Content Accessibility Guidelines (WCAG) ... 254
Webpage slider ... 477
Web tracking ... 366, 576
  profile ... 369
Wholeness ... 261
WH questions ... 175
Wicked problem ... 89
Wireframe ... 254, 436, 438, 487, 506, 536
  BusinessBooster example ... 511
Wish ... 416
Workflow ... 127, 129, 442
  criticality ... 318
  develop ... 430
  main ... 319
Work science ... 36, 37

Y

Year picker ... 463

Service Pages

The following sections contain notes on how you can contact us.

Praise and Criticism

We hope that you enjoyed reading this book. If it met your expectations, please do recommend it. If you think there is room for improvement, please get in touch with the editor of the book: meganf@rheinwerk-publishing.com. We welcome every suggestion for improvement but, of course, also any praise!

You can also share your reading experience via Twitter, Facebook, or email.

Supplements

If there are supplements available (sample code, exercise materials, lists, and so on), they will be provided in your online library and on the web catalog page for this book. You can directly navigate to this page using the following link: http://www.rheinwerk-computing.com/5778. Should we learn about typos that alter the meaning or content errors, we will provide a list with corrections there, too.

Technical Issues

If you experience technical issues with your e-book or e-book account at Rheinwerk Computing, please feel free to contact our reader service: support@rheinwerk-publishing.com.

About Us and Our Program

The website http://www.rheinwerk-computing.com provides detailed and first-hand information on our current publishing program. Here, you can also easily order all of our books and e-books. Information on Rheinwerk Publishing Inc. and additional contact options can also be found at http://www.rheinwerk-computing.com.

Legal Notes

This section contains the detailed and legally binding usage conditions for this e-book.

Copyright Note

This publication is protected by copyright in its entirety. All usage and exploitation rights are reserved by the author and Rheinwerk Publishing; in particular the right of reproduction and the right of distribution, be it in printed or electronic form.

Your Rights as a User

You are entitled to use this e-book for personal purposes only. In particular, you may print the e-book for personal use or copy it as long as you store this copy on a device that is solely and personally used by yourself. You are not entitled to any other usage or exploitation.

In particular, it is not permitted to forward electronic or printed copies to third parties. Furthermore, it is not permitted to distribute the e-book on the Internet, in intranets, or in any other way or make it available to third parties. Any public exhibition, other publication, or any reproduction of the e-book beyond personal use are expressly prohibited. The aforementioned does not only apply to the e-book in its entirety but also to parts thereof (e.g., charts, pictures, tables, sections of text).

Copyright notes, brands, and other legal reservations as well as the digital watermark may not be removed from the e-book.

Digital Watermark

This e-book copy contains a **digital watermark**, a signature that indicates which person may use this copy. If you, dear reader, are not this person, you are violating the copyright. So please refrain from using this e-book and inform us about this violation. A brief email to info@rheinwerk-publishing.com is sufficient. Thank you!

Trademarks

The common names, trade names, descriptions of goods, and so on used in this publication may be trademarks without special identification and subject to legal regulations as such.

All products mentioned in this book are registered or unregistered trademarks of their respective companies.

Limitation of Liability

Regardless of the care that has been taken in creating texts, figures, and programs, neither the publisher nor the author, editor, or translator assume any legal responsibility or any liability for possible errors and their consequences.

Method	Identify user groups	Get to know general requirements	Understand social interactions	Understand user needs and tasks	Evaluate the importance of needs and tasks	Ask for opinions	Present knowledge about users	Get to know the context of use	Get to know processes	Develop understanding of processes	Present knowledge about processes	Evaluation of existing solutions	Reveal problems in the use of solutions	Discover opportunities for further development	Understand first impressions	Test the usability of a solution	Test the user experience of a solution	Choose between different (design) variants	Assess the quality of your own solution	Test the effect of the overall design	Understand how attention is distributed across a design	Capture changes in usage behavior over time	Capture frequencies of use	Capture durations of use	Uncover patterns in solution usage	Optimize solution structure for users
Secondary research/ desk research	◒	●	○	○	○	○	○	◒	○	○	○	◒	◒	○	○	○	○	○	○	○	○	○	◒	◒	○	○
Field observation	◒	●	●	◒	○	○	○	●	●	◒	○	○	●	●	○	◒	◒	○	◒	○	○	●	●	●	●	○
Interview/ jobs-to-be-done	●	●	◒	●	●	●	○	◒	◒	◒	○	●	◒	●	○	○	◒	○	◒	●	○	◒	◒	○	◒	○
Group discussions, focus groups	●	●	◒	●	●	●	○	◒	◒	◒	○	●	◒	●	○	○	◒	○	◒	●	○	◒	◒	○	◒	○
Quantitative survey/ questionnaires	◒	●	○	◒	●	●	○	○	○	○	○	●	◒	◒	○	◒	◒	●	◒	●	○	◒	◒	○	◒	○
Contextual inquiry	●	●	●	●	●	●	○	●	●	●	○	●	●	●	○	○	●	○	◒	○	○	●	◒	◒	●	○
Analysis of quantitative usage data	○	○	○	○	○	○	○	○	◒	○	○	○	◒	◒	○	○	○	○	◒	○	○	●	●	●	●	○
Web tracking/ click analysis	○	○	○	○	○	○	○	○	◒	○	○	○	◒	◒	○	○	○	○	◒	○	○	●	●	●	●	◒
Ticket and feedback analysis	○	●	○	◒	●	○	○	○	○	○	○	◒	●	●	○	○	◒	○	◒	○	○	○	○	○	◒	○
Task analysis	○	●	○	○	○	○	○	○	●	◒	●	○	○	◒	○	○	○	○	○	○	○	○	○	○	◒	◒
User or customer journey mapping/ touchpoint analysis	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	◒	◒
Personas and empathy maps	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Usability and user experience test	○	◒	○	◒	◒	○	○	○	◒	◒	○	●	●	●	●	●	●	◒	●	●	○	○	○	◒	○	●
Usability and user experience sprint	○	◒	○	◒	◒	○	○	○	◒	◒	○	●	●	●	●	●	●	◒	◒	●	○	○	○	◒	○	●
Rapid user tests	○	○	○	○	○	○	○	○	◒	◒	○	●	●	●	●	●	●	○	◒	●	○	○	○	◒	○	●
A/B testing	○	○	○	○	○	○	○	○	○	○	○	●	◒	●	○	◒	◒	●	●	◒	○	○	○	○	◒	●
UX benchmarking	○	○	○	○	●	○	○	○	◒	◒	○	●	●	●	●	●	●	●	●	●	○	○	○	◒	○	●
5-second test	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○	○	◒	○	○	●	○	○	○	○	○
Standardized questionnaires	○	○	○	○	●	●	○	○	○	○	○	●	◒	◒	○	◒	◒	●	●	●	○	○	◒	○	○	○
Diary study	○	◒	◒	◒	◒	●	○	◒	◒	◒	○	●	◒	◒	○	◒	●	○	◒	◒	○	●	◒	◒	◒	○
Card sorting	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
Expert review	○	○	○	○	○	○	○	○	○	○	○	●	◒	◒	○	◒	○	○	◒	○	○	○	○	○	○	○
Association space test (ARTTest)	○	○	○	○	○	○	○	○	○	○	○	◒	○	○	○	○	◒	●	○	●	○	○	○	○	○	○
Eye tracking	○	○	○	○	○	○	○	○	◒	○	○	○	◒	○	○	○	○	◒	○	◒	●	○	○	●	○	○
Physiological measurement methods	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	◒	○	○	○	○	○	○

● = suitable ◒ = partly suitable ○= not suitable